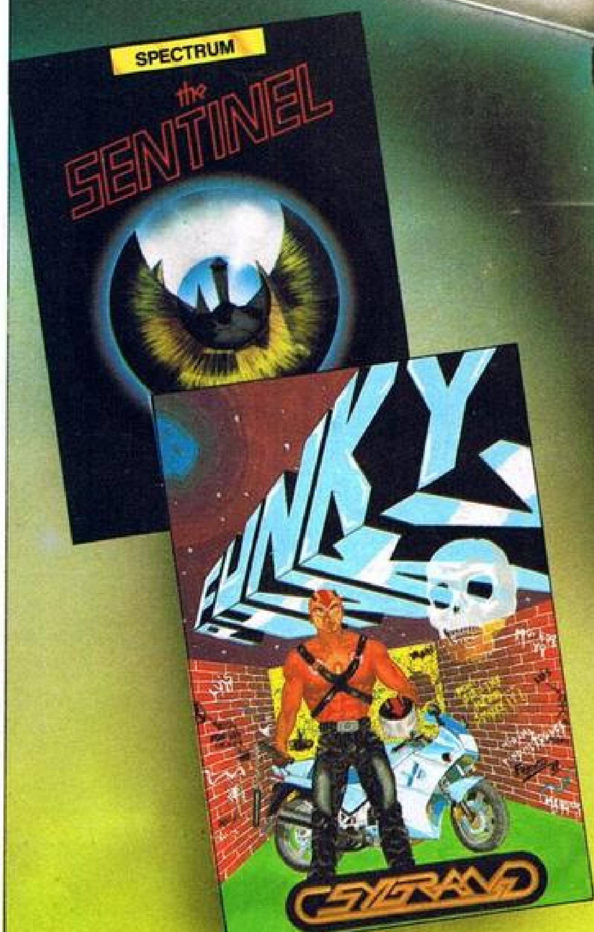
Mundo Spectrum

LOS MEJORES LISTADOS PARA TU MICRO

EDITA SYGRANSA -año I-Nº 2 PVP 200 Pts. IVA Incl.

DRUID

La interpretación de un programa BASIC



LISTADOS:

XADON FORCE

GOL

FLYK

FICHERO

BI - PALETA

MUROS GALACTICOS

PVP CANARIAS 200 Pts.

Mundo Spectrum

Director:
Angel Herrero Fernández
Director Técnico:
Luis Sanguino
Coordinador Editorial:
Félix Santamaría
Software:
Angel García
Secretaria Redacción:
Mercedes Matons
Diseño e Ilustraciones:
Javier Caballero
Colaboradores:
Antonio García
A. Gustavo Chico
Andrés M. García

Mundo SPECTRUM es una publicación del Grupo Editorial SYGRAN, S.A. Polig. Ind. Valdonaire. C/Apolonio Hernández. HUMANES (Madrid).
Director Adjunto:
Andrés Franco

Publicidad y suscripciones:
GENESIS
Tomás López, 3-6°
28009 Madrid
Tel. 401 77 54

Fotocomposición:
Speed Letra, S.A.
Imprime:
Gráficas Osiris.
Brañuelas, 29
Fuenlabrada
Distribuye:
G.M.E. Pza. Castilla, 3.
Madrid
Depósito Legal:
M-31674-1987

SUMARIO

LA INTERPRETACION DE UN PROGRAMA BASIC



Como de sobra se sabe, el lenguaje que usan los ordenadores el tan nombrado y tan desconocido por muchos Código Máquina, es decir, una sucesión interminable de ceros y unos (los bits). Entonces, ¿por qué cuando uno se sienta ante el ordenador y empieza a programar, usa palabras (eso sí, inglesas) que nada tienen que ver con ceros y unos? Lo que está claro es que el ordenador comprende y ejecuta lo que hacemos (siempre que nos atengamos a la sintaxis del BASIC, por supuesto). La clave de esta aparente paradoja está en la inalterable ROM; a los 16 kb que tiene, reside un complicado programa desarrollado por los técnicos de Sinclair, que tiene la difícil tarea de traducirle al microprocesador del Spectrum (el famoso Z-80A) lo que escribe un "impreciso" y variable ser humano: es el denominado INTERPRETE DE BASIC.

Como buen "intérprete" se encarga de traducirle al sufrido microprocesador cada una de las líneas e instrucciones BASIC que nosotros hemos programado, a puro código binario en el que el Z-80 se mueve como pez en el agua.

Para analizar la labor del intérprete hemos de ver previamente cómo se almacena un programa BASIC en la memoria del Spectrum. Lo que uno escribe directamente sobre el teclado no se almacena tal que así, sino que sufre ciertas modificaciones para mayor comodidad del intérprete. En primer lugar no es lo mismo introducir una orden directa (sin número de línea), que pasa inmediatamente a ser traducida y ejecutada, que introducir un programa con sus números de línea correspondientes. Este programa queda almacenado en RAM a partir de la dirección contenida en la variable del sistema PROG, situada en las posiciones 5C53H y 5C54H. Supongamos que escribimos el siguiente programa:

```
23755:  0
23756:  10
23757:  8
23758:  0
23759:  234     REM
23760:  80      P
23761:  82      R
23762:  85      U
23763:  69      E
23764:  66      B
23765:  65      A
23766:  13
23767:  0
23768:  15
23769:  97      a
23770:  0
23771:  241     LET
23772:  80      P
23773:  82      R
23774:  79      O
23775:  71      G
23776:  61      =
23777:  40      (
```

```
23778:  190     PEEK
23779:  50      2
23780:  51      3
23781:  54      6
23782:  51      3
23783:  53      5
23784:  14
23785:  0
23786:  0
23787:  83      S
23788:  92      \
23789:  0
23790:  43      +
23791:  50      2
23792:  53      5
23793:  54      6
23794:  14
23795:  0
23796:  0
23797:  0
23798:  1
23799:  0
23800:  42      *
23801:  190     PEEK
23802:  50      2
23803:  51      3
23804:  54      6
23805:  51      3
23806:  54      6
23807:  14
23808:  0
23809:  0
23810:  84      T
23811:  92      \
23812:  0
23813:  41      )
23814:  58      :
23815:  241      LET
23816:  70      F
23817:  80      P
23818:  82      R
23819:  79      O
23820:  71      G
23821:  61      =
23822:  40      (
23823:  190     PEEK
23824:  50      2
23825:  51      3
23826:  54      6
23827:  50      2
23828:  55      7
23829:  14
23830:  0
23831:  0
23832:  75      K
23833:  92      \
23834:  0
23835:  43      +
23836:  50      2
23837:  53      5
23838:  54      6
23839:  14
23840:  0
23841:  0
23842:  0
23843:  1
23844:  0
23845:  42      *
23846:  190     PEEK
23847:  50      2
23848:  51      3
23849:  54      6
23850:  50      2
23851:  56      8
23852:  14
23853:  0
23854:  0
23855:  76      L
23856:  92      \
23857:  0
23858:  41      )
23859:  45      -
23860:  49      1
23861:  14
23862:  0
23863:  0
23864:  1
23865:  0
23866:  0
23867:  13
23868:  0
23869:  20
23870:  14
23871:  0
23872:  235      FOR
23873:  73      I
23874:  61      =
23875:  80      P
23876:  82      R
23877:  79      O
23878:  71      G
23879:  204      TO
23880:  70      F
23881:  80      P
23882:  82      R
23883:  79      O
23884:  71      G
23885:  13
23886:  0
23887:  30
23888:  39
23889:  0
23890:  224      LPRINT
23891:  73      I
23892:  59      ;
23893:  34      "
23894:  58      :
23895:  32
23896:  34      "
23897:  59      ;
23898:  190     PEEK
23899:  73      I
23900:  59      ;
23901:  173     TAB
23902:  49      1
23903:  53      5
23904:  14
23905:  0
23906:  0
23907:  15
23908:  0
23909:  0
23910:  59      ;
23911:  40      (
23912:  194     CHR$
23913:  190     PEEK
23914:  73      I
23915:  198      AND
23916:  190     PEEK
23917:  73      I
23918:  62      >
23919:  51      3
23920:  49      1
23921:  14
23922:  0
23923:  0
23924:  31
23925:  0
23926:  0
23927:  41      )
23928:  13
23929:  0
23930:  40      (
23931:  3
23932:  0
23933:  243      NEXT
23934:  73      I
23935:  13
```

Si lo ponemos en marcha observamos algo así:

```
10 REM PRUEBA
15 LET PROG=(PEEK 23635+256*PEEK
 23636): LET FPROG=(PEEK 23627+2
56*PEEK 23628)-1
20 FOR I=PROG TO FPROG
30 LPRINT I;":";PEEK I;TAB 15;(C
HR$ PEEK I AND PEEK I>31)
40 NEXT I
```

Los dos primeros bytes vemos que son 0 y 10 que, lógicamente, indican el número de línea. Pero ojo, el primer byte es el de mayor peso, al contrario de como se almacenan las variables del sistema. Viendo esto se podría pensar que el mayor número de línea posible sería 65.535 (poniendo a 255 los dos bytes) pero el intérprete sólo admite números de línea iguales o inferiores a 9.999. Si POKEamos en esos bytes para dar un número de línea mayor podemos conseguir efectos muy curiosos.

Los dos bytes siguientes (8 y 0) almacenan la longitud total de la línea más el código del carácter ENTER (13). Ahora el byte de mayor peso es el segundo, lo que indica que la línea 10 tiene 8 bytes de longitud, (7 si no contamos el ENTER). A continuación vemos el número 234, que corresponde al código del comando REM y, seguidamente, los códigos ASCII de cada una de las letras de la palabra PRUEBA. La línea acaba con el número 13, que es el código de ENTER. Como se ve, las líneas se almacenan de una manera bastante sencilla. Sólo hay una salvedad, pues la manera de almacenar los números es un tanto peculiar. Se podría pensar que cuando se introduce el número 2, éste se almacenaría en memoria con el valor 2 o, quizás, con el valor de su código ASCII (50). Pero en realidad es un poco más complicado. Los números se almacenan en un formato llamado coma flotante. Cuando el Editor de la ROM se encuentra un número lo primero que hace es almacenar su código ASCII; a continuación almacenan el código 14, que es el que indica al intérprete que lo que viene a continuación es la representación del número en coma flotante, que consta de 5 bytes distribuidos de la siguiente manera:

- 1 byte de exponente.
- 1 byte de signo.
- 31 byte de mantisa (4 bytes - 1 bit).

Así, un número ocupa en memoria tantos bytes como cifras tenga más 6 bytes.

Sin embargo, los números que forman parte de una cadena (es decir, que van entre comillas) se almacenan nada más que con su código ASCII, pues el ordenador los trata como simples caracteres.

Después del programa BASIC viene una zona indicada por la dirección contenida en la variable del sistema VARS (5C4B H y 5C4C H) en la que están contenidas todas las variables del programa BASIC. Una vez que hemos visto la forma de guardarse un programa BASIC. en memoria, estamos listos para saber lo que ocurre cuando se pulsa la tecla RUN y el programa empieza a funcionar.

Todas las rutinas residentes en la ROM son llamadas desde una rutina principal (MAIN) que ejecuta un bucle constante. Esta rutina consta de diversas partes, cada una de ellas con un cometido distinto:

- MAIN-EXEC. Se encarga de hacer un listado automático.
- MAIN-1. Limpia todas las zonas de trabajo.
- MAIN-2. Produce la apertura del canal K (teclado) y llama a la rutina del editor.
- MAIN-3. Ejecuta una línea o comando directo.
- MAIN-4. Indica la dirección de retorno de la ejecución de un comando o programa.
- MAIN-5 a MAIN-9. Ajusta las variables del sistema SUBPPC, OLDPPC y OSPPC
- REP-MESS. Esta es la tabla donde están almacenados los mensajes de error del BASIC.
- MAIN-ADD. Es la rutina encargada de sustituir o añadir una línea nueva en el listado.

Todas estas rutinas actúan en forma de bucle continuo. Cuando MAIN-2 llama al EDITOR, este espera a que se pulse alguna tecla y va almacenando lo que se introduce en el espacio de trabajo (esto ocurre sólo cuando es llamado por una sentencia INPUT) o en el área de edición si lo que se escribe es una línea BASIC o un comando directo. En nuestro caso, cuando pulsamos la tecla RUN y damos al ENTER, el EDITOR deja el código de RUN (247) en el área BASIC y retorna a MAIN-2. MAIN-2 cede el control a MAIN-3, que pasa a llamar a una rutina denominada LINE-RUN cuya misión es ejecutar una instrucción o programa. Al detectar el código de RUN llama a la dirección 1COD H, que es una rutina que busca una expresión numérica a continuación del comando y, en caso de no existir parámetros insertar el valor cero por defecto (por eso RUN en realidad RUN 0). Cuando se regresa de esta llamada LINE-RUN por fin llama a la dirección 1EA1 H, o sea, la propiamente dicha rutina del comando RUN, encargada de ejecutar un programa BASIC.

En realidad, esta rutina consta de una especie de bucle que chequea las líneas del programa y va llamando a las rutinas de cada comando o función que encuentra. Así, cuando detecta el comando REM automáticamente salta a la línea siguiente; cuando detecta en la línea 15 el código 241 (LET) pasa a ejecutar las rutinas que localizan una variable y le asignan un valor (en nuestro ejemplo PROG = PEEK...). Y así sucesivamente hasta que se ejecuta la última instrucción de la última línea del programa BASIC. En ese momento, y si el programa no ha hecho algo raro (POKEado en memoria por ejemplo) el control vuelve a MAIN que nuevamente comenzará a enredarse en un bucle en espera de una nueva orden del amo omnipotente (el dueño del ordenador claro).

Como se puede apreciar, este artículo no ha pretendido dar una descripción completa y paso a paso de la difícil tarea que lleva a cabo el intérprete de la ROM, sino ir introduciendo al lector en lo que en sucesivos artículos será un completo y fascinante viaje a través de las calenturientas y eficientes rutinas "ROMianas".

NOTA: Para finalizar sugiero que, a modo de experimento, sean sustituidos todos los valores numéricos del programa ejemplo por sus equivalentes cadenas alfanuméricas. Me explico: si tenemos la línea:

```
15 LET PROG=(PEEK 23635+256*PEEK...
```

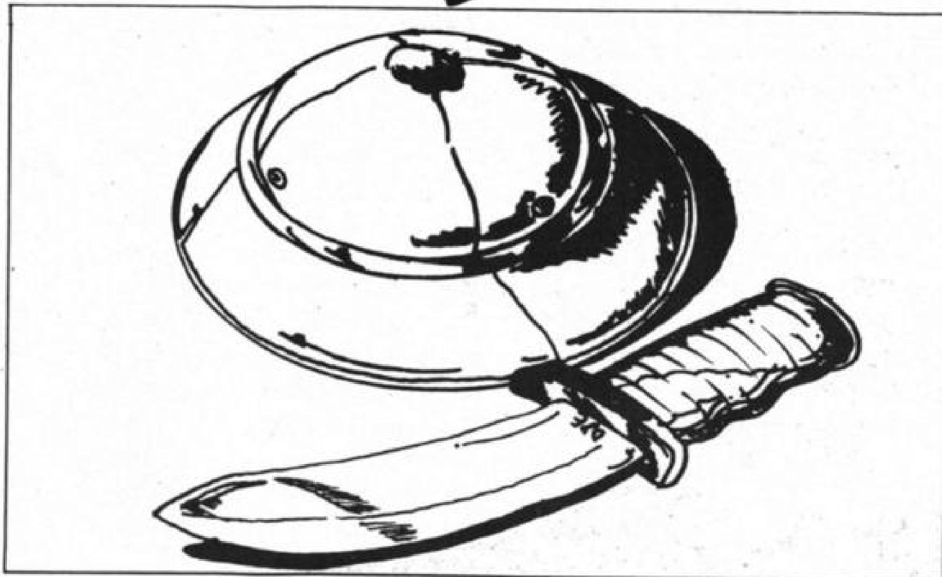
cambiémosla por:

```
15 LET PROG=(PEEK VAL "23635"+VAL"256"*PEEK...
```

Al ejecutarlo se verán algunas diferencias en la forma de almacenarse ahora.



JACK the nipper



- ***JACK THE NIPPER II IN COCONUT CAPERS***
- ***GREMLIN GRAPHICS***
- ***DISTRIBUIDO POR ERBE***

Las clases han acabado y los padres de Jack the Nipper, deciden ir de vacaciones a Australia, a ver a la familia. Pero un viaje de tal calibre, no agrada a nadie, y menos si ese alguien es "Jack the Nipper".

Por tanto la solución más sencilla, es hacer una escala paracaidista en pleno Africa, en el pueblo de GAFONES.

Y aquí es donde comienza la epopeya, culebras, águilas, elefantes, abejas y hasta el mismisimo TARZAN, que no van a consentir que un mocoso como Jack, perturbe la tranquilidad de la selva.

A Jack the Nipper ya solo le quedan dos opciones;

• Morir en la selva a manos de sus enemigos.

• ó encontrar el templo del "HI NAUGHYNESS" y desvelar el misterio que sobre este se cierne.

A través de nuestro recorrido por la selva, podremos encontrar unos u otros objetos que nos serán de utilidad;

• COCOS; AL cogerlos nos darán 17 disparos, que nos permitirán acabar con nuestros enemigos.





ESCUDOS; Seremos inmunes durante un tiempo limitado.

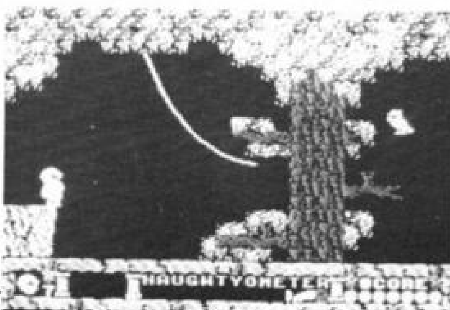
CARRETILLA; Podremos transportarnos por sitios de otra forma inaccesibles. Con los controles podremos determinar la velocidad de la carretilla.

Otros objetos de interés, pueden ser; el tarro de miel, el ratón, la cuerda, y otros muchos más, que utilizándolos de la forma correcta pueden hacer subir nuestro NAUGHTYMETER.

También encontraremos los personajes más pintorescos, que intentaran acabar con nosotros; vaqueros, exploradores y nativos.

• Resumiendo; Jack the Nipper II, ha logrado desbancar a su predecesor en calidad y adictividad. De 3D se ha pasado a 2D, las pantallas monocromas han pasado a multicolor, el movimiento se ha agilizado dando mayor rapidez al juego y los efectos musicales son superiores a los de la versión anterior.

Es un juego rentable, que promete horas y horas de diversión, debido a su





entrañable dificultad, y al gran número de sus pantallas.

PARA QUE TE SEA MAS FACIL, PUEDES USAR LOS SIGUIENTES POKES QUE TE OFRECEMOS.

Vidas Infinitas	Poke 43251,0
Objetos Infinitos	Poke 38306,0

(Sirve para cocos y escudos)

CARGADOR

```
10 CLEAR 24575
20 LOAD "" SCREEN$
30 LOAD "" CODE
40 POKE 43251,0: REM vidas infinitas.
50 POKE 38306,0: REM objetos infinitos.
60 RUN USR 34240
```

Si no quieres VIDAS INFINITAS al copiar este cargador no copies la línea 40.

Si no quieres OBJETOS INFINITOS al copiar el cargador no copies la línea 50.

Después de teclearlo pulsa RUN (Enter) y carga el original desde el principio.

DISCIPLE

En nuestro primer número hablamos del Disciple, un interfaz que permite conectar al Spectrum dos unidades de disco, impresora con interfaz Centronics y dos mandos de juegos.

Pero ha llegado a nuestras manos la versión 3B de este dispositivo, que amplía las posibilidades en lo referente al disco y la impresora. Estas nuevas posibilidades son:

1. Salvar un programa de 128K.
2. Salvar a disco una pantalla.
3. Hacer COPY de la pantalla a impresora, en doble tamaño y con interpretación de colores por trama de grises. (Sólo con impresoras compatibles EPSON).
4. Generación de ficheros ejecutables.
5. Acceso aleatorio a disco, que nos permite leer y escribir sectores individuales.
6. Posibilidad de ampliar ficheros existentes. Desde BASIC podemos abrir y cerrar corrientes y canales de los ficheros secuenciales.
7. En la "Red Local" es posible forrar un fichero sobre una estación.

Muchas de las posibilidades mencionadas son accesibles pulsando la tecla SHIFT Spectrum y a continuación el botón SNAPSHOT. Haciendo esto, el border permanecerá parpadeante en espera de la pulsación de una de las opciones del siguiente menú imaginario:

— Copy de la pantalla en impresora a tamaño normal. Hay que hacer notar que la copia es interpretada. Los colores 0-3 son impresos de tinta y los colores 4-7 conservan el color del papel.

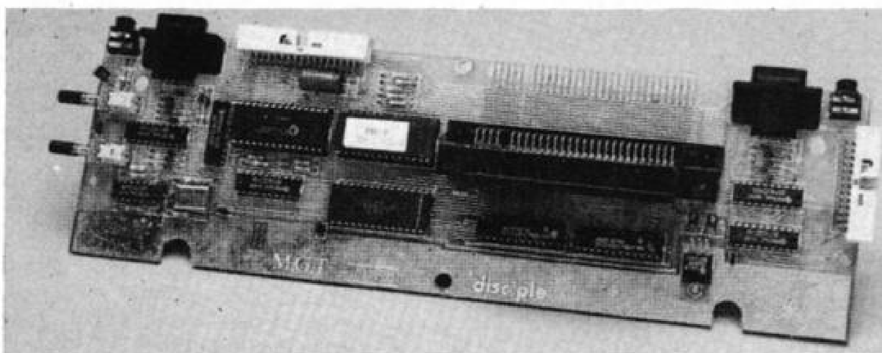
— Copy de la pantalla a doble tamaño. Los colores son interpretados con tramas de grises.

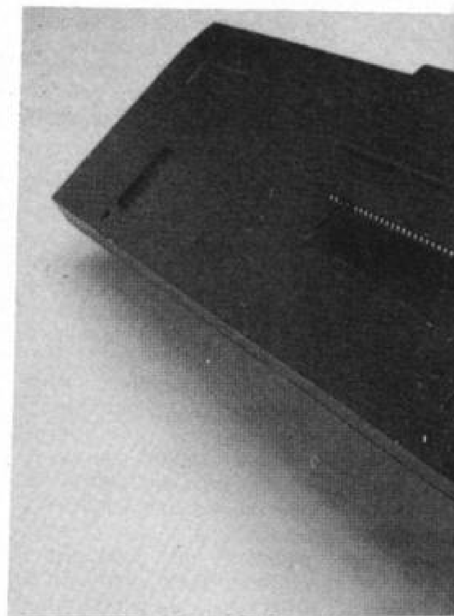
— Graba la pantalla (SCREEN) a disco.

— Graba 48 K de memoria a disco. El programa grabado continúa su ejecución en el mismo punto donde se dejó. Esto nos permite grabar, por ejemplo, un juego comercial a disco en cualquier punto de su ejecución.

— Graba 128 K de memoria a disco. Igual que la opción anterior, pero para el software escrito en el Spectrum + de 128 K.

El disciple dispone de una RAM propia, por lo cual no consume memoria. En esta RAM es donde reside la configuración del sistema operativo.





POSICION	FUNCION	VALORES NORMALES
0	Determina si parpadea o no el BORDER durante las operaciones de disco. Se realiza un AND del número del sector con el dato contenido en ésta posición.	0 - 7
1	Formato de la unidad de disco 1. Indica el número de pistas del disco y si es de simple o doble cara.	número de pistas + 128* doble cara 40, 80, 168, 208.
2	Formato de la unidad 2	número de pistas + 128* doble cara.
3	Step rate (tiempo entre cada impulso de disco). El valor mínimo es 3, pero el más aconsejable es 6. Normalmente depende de la unidad de disco	3-255
4	Determinar si se emplea o no el interfaz CENTRONICS	0,1
5	Fija la anchura de impresión en caracteres. Similar a WIDTH del Spectrum + 2. Su valor normal es 80.	0-80

DISCIPLE



POSICION	FUNCION	VALORES NORMALES
6	Indicador para la impresora. Dice si ha de filtrar o no los códigos de control. Un cero filtra los códigos de color y expande los TOKENs. Un 1 envía todos los códigos tal cual.	0-1
7	Avance de línea. se da en n/72 de pulgada. Su valor normal es 12.	n/72
8	Avances de línea. Indica el número de líneas en blanco que se dejan después de cada retorno de carro. Para impresoras con avance de carro automático tendremos que darle el color.	0,255
9	Fija margen izquierdo. Se envían n espacios, después de cada retorno de carro.	0,255
10	Indicador que dice si los caracteres @, # y £ han de ser generados como gráficos (valor 1) o simplemente hay que enviar su código (valor 0). Hay que recordar que en los listados los U.D.G.s son impresos en modo gráfico.	0,1
11	Número de estación de la red. Vale 1 para la estación maestra.	1,63

Gracias a que se ha previsto una instrucción para que el BASIC pueda escribir en esta RAM, nos es posible modificar algún punto de la configuración del sistema operativo, sin necesidad de cargar el programa en cinta creado para este fín.

El formato de la instrucción BASIC es el siguiente:

POKE @ dirección, valor

La forma en que el disciple reconoce esta instrucción es similar al del interfaz 1 de Sinclair. Al producirse un error sintáctico, el control pasa al Disciple para comprobar si se trata del mencionado comando.

A continuación, os vamos a listar las posiciones de memoria de la RAM del Disciple con sus funciones y los valores que pueden contener para adaptar el sistema a nuestras necesidades:

TABLA

Existen otros grupos de direcciones en la RAM del DISCIPLE que nos pueden resultar de utilidad:

En las direcciones 18-25 se guarda la secuencia de códigos necesaria para inicializar la impresora, seguido por el código 12, como marca de final.

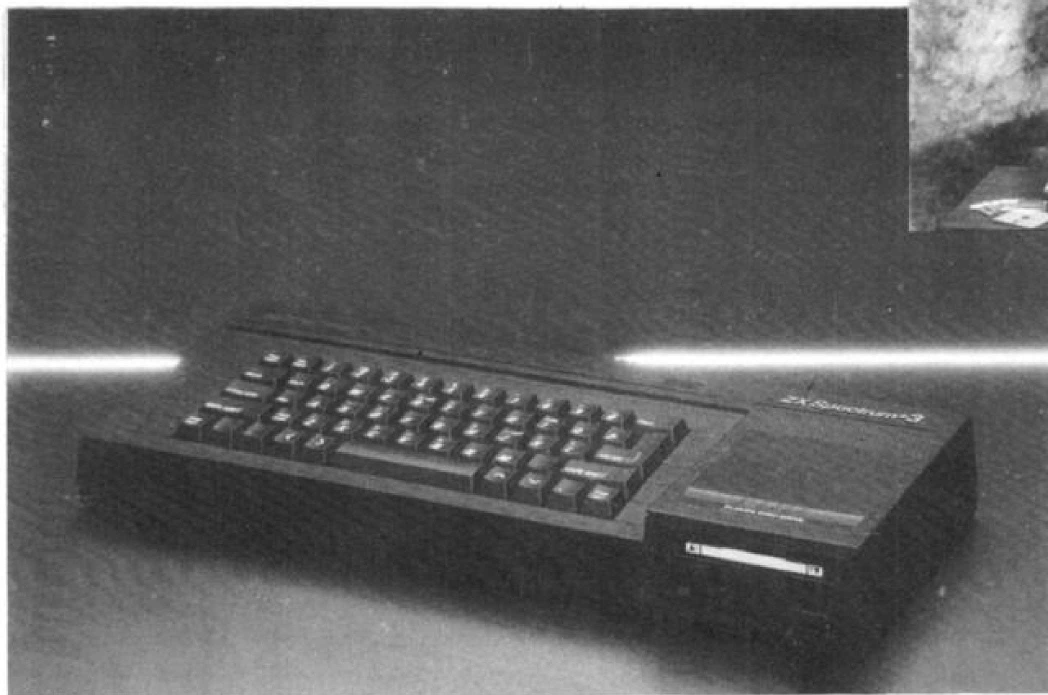
En las posiciones 42-49 se almacenan los códigos que se han de enviar para pasar la impresora a modo gráfico. Igual que en el caso anterior, el final de la secuencia es marcado con el código 128.

Por último, y creemos que da gran interés, son las posiciones 14 y 15 que almacenan la dirección de memoria donde se produce un salto indirecto al aparecer un error de sintaxis. En el caso de contener el valor 0, el sistema operativo de disco devuelve el control a la ROM del, Spectrum. Esto nos permite incorporar nuevos comandos al Sistema Operativo.

A nuestro juicio, el DISCIPLE es un interfaz que resolverá muchos de los problemas de expansión de nuestro Spectrum. La posibilidad de almacenar información en grandes cantidades y a gran velocidad con el manejo de hasta dos unidades de disco y la posibilidad de adaptar cualquier tipo de impresora, lo convierten en un elemento casi indispensable en el hardware que acompaña a nuestro micro.



NUEVO Spectrum+3



Ha sido presentado recientemente en Madrid el, tan anunciado, SINCLAIR PLUS 3 que incorpora importantes mejoras sobre los anteriores equipos de la familia SPECTRUM; entre ellas la más destacable es la unidad de disco de 3 pulgadas muy similar a la de los ordenadores AMSTRAD CPC 664 y 6128 pero incompatible con ellos. La capacidad del disco del SINCLAIR PLUS-3 es de 173 kb formateados desde el BASIC de éste ordenador.

Otras características destacables de éste ordenador son su interface paralelo centronics que permite la conexión de una impresora compatible EPSON, la salida MID RS232 para comunicaciones, dos conectores para JOYSTICKS y una salida RGB para monitor.

La memoria RAM del SINCLAIR PLUS-3 es de 128 kbytes divididos en dos bancos de 64 kb cada uno, pudiendo ser utilizado el segundo como un DISCO RAM.

El BASIC, sin perder la compatibilidad con los modelos anteriores, incorpora también importantes ampliaciones como la instrucción de sonido PLAY que permite realizar melodías mediante los tres canales y ocho octavas, un nuevo CHIP de sonido independiente del procesador central.

Otras características de éste BASIC son sus comandos para la utilización del disco (formateo, gestión de ficheros, etc.), comandos para acceder al RAM DISC (disco simulado en memoria), posibilidad de evitar el filtro de caracteres para evitar códigos a impresora, etc.

El SINCLAIR PLUS-3 permite también la utilización del cassette mediante un conector JACK SETERO, así como la conexión a TV o monitor RGB mediante dos salidas independientes.

Esperamos poder ampliar ésta noticia en próximos números de MUNDO SPECTRUM, con la realización de un exahustivo banco de pruebas.

INDESCOMP PASA A MANOS DE AMSTRAD



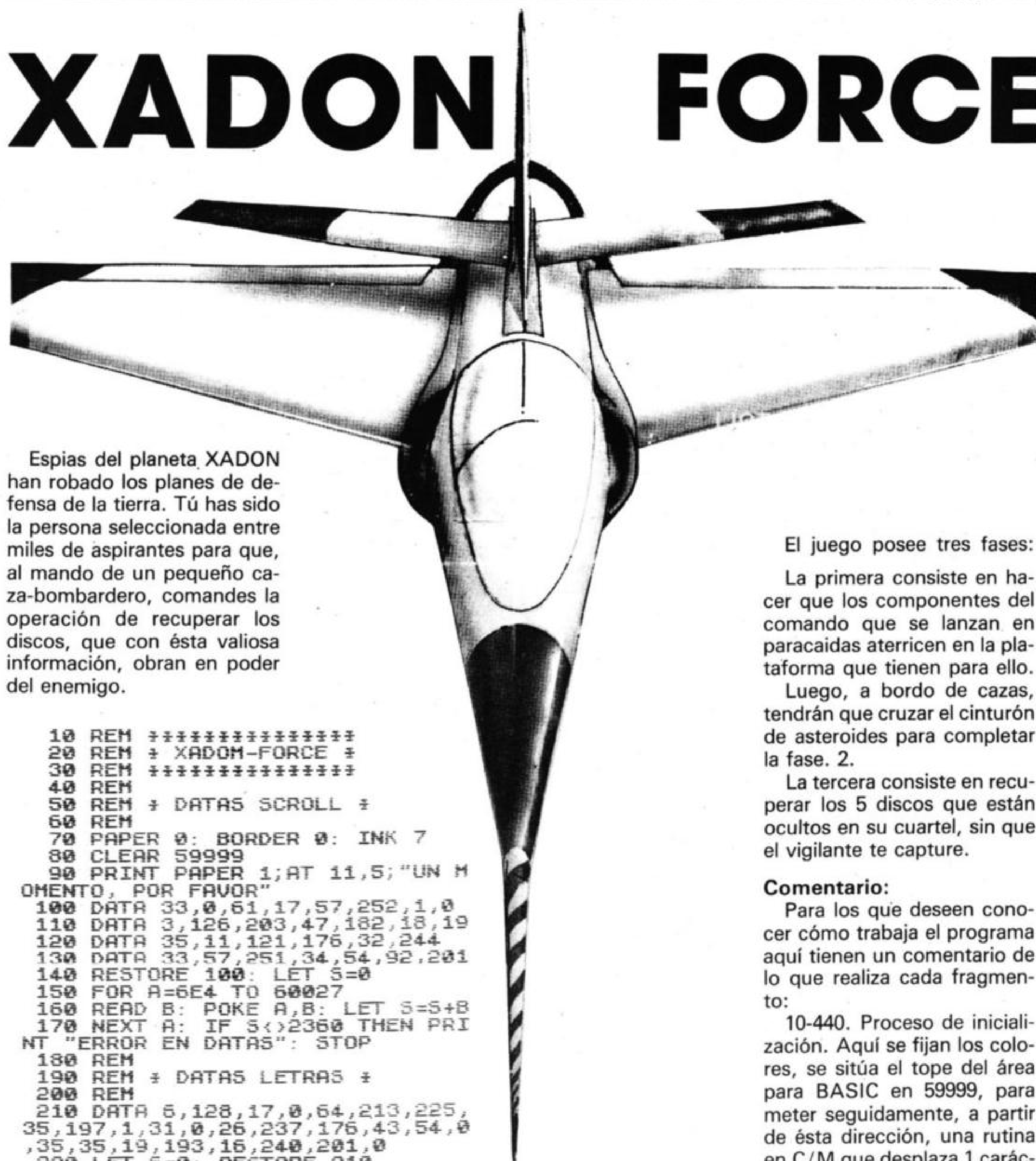
Durante la celebración del 25 certamen de SONIMAG en Barcelona en los últimos días del mes de septiembre, fue dada a conocer en rueda de prensa a los medios informativos la noticia de la compra del 100 % de las acciones del GRUPO INDESCOMP por parte de la firma inglesa AMSTRAD PLC de cuyos productos, INDESCOMP es distribuidor exclusivo en nuestro país.

Alan Sugar, presidente de AMSTRAD PLC, se desplazó hasta Barcelona para dar a conocer la noticia de la operación que supone la absorción por parte de la compañía inglesa del total de la plantilla de Indescomp.

Alan Sugar informó de su propósito de que en un plazo breve fueran construídos en España algunos de los componentes de los equipos AMSTRAD, lo cual daría un fuerte impulso a la empresa.

El acuerdo, pendiente de aprobación por parte del organismo competente del gobierno español, será primado en breve y el coste total de la operación podría alcanzar los 250 millones de libras esterlinas: 216 a la firma del contrato y el resto, en función de los beneficios obtenidos en los primeros 5 años.

XADON FORCE



Espias del planeta XADON han robado los planes de defensa de la tierra. Tú has sido la persona seleccionada entre miles de aspirantes para que, al mando de un pequeño caza-bombardero, comandes la operación de recuperar los discos, que con ésta valiosa información, obran en poder del enemigo.

```
  10 REM ***************
  20 REM * XADOM-FORCE *
  30 REM ***************
  40 REM
  50 REM * DATAS SCROLL *
  60 REM
  70 PAPER 0: BORDER 0: INK 7
  80 CLEAR 59999
  90 PRINT PAPER 1;AT 11,5;"UN M
OMENTO, POR FAVOR"
 100 DATA 33,0,61,17,57,252,1,0
 110 DATA 3,126,203,47,182,18,19
 120 DATA 35,11,121,176,32,244
 130 DATA 33,57,251,34,54,92,201
 140 RESTORE 100: LET S=0
 150 FOR A=6E4 TO 60027
 160 READ B: POKE A,B: LET S=S+B
 170 NEXT A: IF S<>2360 THEN PRI
NT "ERROR EN DATAS": STOP
 180 REM
 190 REM * DATAS LETRAS *
 200 REM
 210 DATA 6,128,17,0,64,213,225,
35,197,1,31,0,26,237,176,43,54,0
,35,35,19,193,16,240,201,0
 220 LET S=0: RESTORE 210
 230 FOR A=60100 TO 60125
 240 READ B: POKE A,B: LET S=S+B
 250 NEXT A: IF S<>2192·THEN PRI
NT "ERROR EN DATAS": STOP
 260 REM
 270 LET A=USR 60000
 280 POKE 23658,8
 290 REM
 300 REM * DATAS GRAFICOS *
 310 REM
 320 DATA 254,254,254,0,239,239,
239,0
 330 DATA 224,120,60,127,248,252
```

El juego posee tres fases:

La primera consiste en hacer que los componentes del comando que se lanzan en paracaidas aterricen en la plataforma que tienen para ello.

Luego, a bordo de cazas, tendrán que cruzar el cinturón de asteroides para completar la fase. 2.

La tercera consiste en recuperar los 5 discos que están ocultos en su cuartel, sin que el vigilante te capture.

Comentario:

Para los que deseen conocer cómo trabaja el programa aquí tienen un comentario de lo que realiza cada fragmento:

10-440. Proceso de inicialización. Aquí se fijan los colores, se sitúa el tope del área para BASIC en 59999, para meter seguidamente, a partir de ésta dirección, una rutina en C/M que desplaza 1 carácter, a la izquierda el contenido de los 2/3 superiores de la pantalla y otra rutina que genera en la parte alta de la RAM, otro banco gráfico donde las letras son el doble de gruesas que las del banco original.

El proceso que sigue es tomar cada uno de los bytes

que definen los caracteres, lo desplaza 1 bit a la derecha y se realiza un OR con el dato inicial, de ésta forma tenemos un carácter con el doble de grosor. La rutina lleva una comprobación que en caso de introducir mal las líneas DATA te lo indicará. Si ocurre, repasa las líneas 100-130. Seguidamente, en las líneas 300-400 se cambian las definiciones de las letras a-j por las de nuestros gráficos.

Esto se ha hecho de ésta manera para poder distinguir y reconocer cada gráfico en pantalla usando la instrucción SCREEN$ (x, y). Si hubiésemos utilizado los UDGs normales ésta instrucción nos devolvería siempre la cadena vacía, en vez del código del carácter que nos proporciona el método usado.

Al introducir los gráficos se realiza otro "CHECKSUM", que en caso de informar de un error tendrás que repasar las líneas 320-410.



450-620. Presentación. Se recupera en la variable R el valor del record, almacenado en las posiciones de memoria 60200-1. Tras un efecto sonoro se imprime el título contenido en una ventana.

Más tarde se imprime el resto del texto, empleando una subrutina de centrado. Se espera la pulsación de una tecla para dar paso al juego. La espera se realiza con PAUSE 0, precedido de un PAUSE 15 para anular cualquier pulsación involuntaria previa. dibuja el escenario, entrando

```
,127,63
 340 DATA 0,0,0,0,192,60,255,240
 350 DATA 52,126,255,127,246,255
,110,60
 360 DATA 60,36,60,153,126,60,36
,102
 370 DATA 255,153,187,153,255,16
5,129,255
 380 DATA 60,66,189,181,165,189,
66,60
 390 DATA 255,255,255,231,231,25
5,248,255
 400 DATA 34,8,66,16,9,80,2,16
 410 DATA 60,94,191,191,255,129,
66,36
 420 LET S=0: RESTORE 320
 430 FOR A=0 TO 79: READ B: POKE
 65089+A,B: LET S=S+B: NEXT A
 440 IF S<>11077 THEN PRINT "ERR
OR EN DATAS": STOP
 450 REM
 460 REM * PRESENTACION *
 470 REM
 480 LET R=PEEK 60200+256*PEEK 6
0201
 490 FOR A=0 TO 60 STEP 2
 500 BEEP .01,A: NEXT A
 510 CLS
 520 INK 2: PAPER 7: PRINT "aaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa"
 530 FOR A=1 TO 5: PRINT AT A,0;
"a
a": NEXT A
 540 PRINT "aaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaa"
 550 PRINT FLASH 1; INK 0;AT 3,9
;" XADOM-FORCE "
 560 PAPER 0: INK 7
 570 PRINT AT 9,0: LET A$="1987
MUNDO SPECTRUM": GO SUB 2030: LE
T A$="AUTOR: ANGEL GARCIA DELGAD
O": GO SUB 2030: LET A$="CONTROL
ES: Q.A.O.P.SPACE": GO SUB 2030:
 LET A$="PULSE UNA TECLA PARA SE
GUIR": GO SUB 2030: PAUSE 15: PA
USE 0: BEEP .01,30
 580 REM
 590 CLS
 600 LET P=0
 610 LET V=0
 620 LET VV=5
 630 REM
 640 REM * FASE 1 *
 650 REM
 660 REM * PAISAJE *
 670 REM
 680 LET F=1: GO SUB 2130
 690 LET VV=VV-1: IF VV=0 THEN G
O TO 990
 700 CLS : PRINT PAPER 4;AT 17,0
;"iiiiiiiiiiiiiiiiiiiiiiiiiiiiii
iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii
ii"
 710 GO SUB 2070
 720 PRINT FLASH 1; PAPER 1; INK
 7;AT 17,15;"▛▜"
 730 FOR A=0 TO 255 STEP 8: PLOT
 INK RND*7;A,50+RND*125: NEXT A
 740 REM
 750 REM * INICIALIZA *
 760 REM
 770 LET NX=0: LET NA=0: LET PX=
0: LET PY=0
 780 REM
 790 REM * BUCLE PRINCIPAL *
 800 REM
```

```
 810 PRINT AT 2,NX;"   ": LET NX
=NX+1: IF NX>29 THEN LET NX=0
 820 PRINT AT 2,NX; INK RND*7;"i
"; INK 5;"bc"
 830 BEEP .001,10: BEEP .001,20
 840 IF NA=0 THEN IF RND>.9 THEN
 IF NX>4 AND NX<26 THEN LET NA=1
: LET PX=3: LET PY=NX+1
 850 IF NA=0 THEN BEEP .01,15: G
O TO 810
 860 PRINT AT PX,PY;" ";AT PX+1,
PY;" "
 870 IF INKEY$="O" THEN LET PY=P
Y-1: IF PY<1 THEN LET PY=30
 880 IF INKEY$="P" THEN LET PY=P
Y+1: IF PY>30 THEN LET PY=1
 890 LET PX=PX+1
 900 PRINT INK 3;AT PX,PY;"j"; I
NK 7;AT PX+1,PY;"e";
 910 IF PX>14 THEN GO TO 960
 920 GO TO 810
 930 REM
 940 REM * ATERRIZAJE *
 950 REM
 960 IF PY=15 OR PY=16 OR PY=17
THEN LET P=P+125: PRINT AT 15,PY
;" ": LET V=V+1: BEEP .05,5: BEE
P .1,25: PAUSE 5: PAUSE 60: GO T
O 690
 970 FOR B=1 TO 5: FOR A=1 TO 7:
 PRINT INK A;AT PX+1,PY-1;"iii";
AT PX,PY-1;"iii": BEEP .01,A: NE
XT A: NEXT B
 980 PAUSE 5: PAUSE 60: GO TO 69
0
 990 REM
1000 REM * FASE 2 *
1010 REM
1020 IF V=0 THEN GO TO 2230
1030 LET F=2: GO SUB 2130
1040 REM
1050 REM * PAISAJE *
1060 REM
1070 CLS : GO SUB 2070: PRINT IN
K 5;AT 16,0;"gggggggggggggggggg
gggggggggggggg";: FOR A=1 TO 2: P
RINT INK 2;"aaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaa": NEXT A
1080 PRINT AT 0,0;: FOR A=3 TO 1
 STEP -1: PRINT PAPER A,,,,,,,,:
 NEXT A: PAPER 8
1090 REM
1100 REM * INICIALIZA *
1110 REM
1120 LET NX=8: LET C=0
1130 REM
1140 REM * BUCLE PRINCIPAL *
1150 REM
1160 PRINT AT NX,4;"   "
1170 IF INKEY$="Q" THEN LET NX=N
X-1: IF NX<1 THEN GO TO 1300
1180 IF INKEY$="A" THEN LET NX=N
X+1: IF NX>15 THEN GO TO 1300
1190 IF SCREEN$ (NX,7)="d" THEN
GO TO 1300
1200 IF SCREEN$ (NX,5)="d" THEN
GO TO 1300
1210 IF SCREEN$ (NX,6)="d" THEN
GO TO 1300
1220 PRINT AT NX,5;"ibc": IF C<1
80 THEN PRINT AT RND*15,30;"d"
1230 BEEP .003,10: BEEP .003,15:
 BEEP .003,20
1240 LET P=P+5: LET C=C+1: IF C>
200 THEN GO TO 1340
```

así en un bucle que se va a repetir VV veces, y que va a servir para conseguir las vidas a usar durante el juego.

En la línea 730 se dibujan las estrellas.

En las líneas 790-920 se encuentra el bucle principal de esta fase, donde se hace lo siguiente:

— Borramos nuestra nave e incrementamos su coordenada. A continuación se imprime.

— Si no hay paracaidista, se activa según una condición al azar para que exista, inicializando sus coordenadas.

— En la línea 850 comprobamos de nuevo si existe el paracaidista. Si es así, se mueve según la acción de las

Por último antes de pasar a la fase 1, se inicializa P (puntos) a 0, V (vidas) a 0 y VV (vidas posibles) con 5.

630-980. Fase 1. Se informa al jugador de la fase en la que está. A continuación se



teclas, si no vuelve a repetir el bucle.

— Si el paracaidista supera la coordenada x = 14, entonces se pasa a comprobar si está en la plataforma o fuera de ella, actuándose en consecuencia. En el primer caso, suma puntos y se pasa al siguiente. En el segundo caso, se imprime una explosión y se continúa con el siguiente hasta que no queden más.

Si las vidas son mayores que cero se pasa a la FASE 2.

990-1330. FASE 2: Se informa que estamos en la fase 2. Luego en las líneas

1050-1090 se dibuja el escenario. A continuación se inicializan dos variables; nuestra coordenada X, llamada NX, y el contador de zona recorrida C.

Entramos así en el bucle principal de ésta fase, que recoge las líneas 1140-1260 donde hacemos lo siguiente:

— Borrar nuestra nave.

— Mover la nave según las teclas pulsadas.

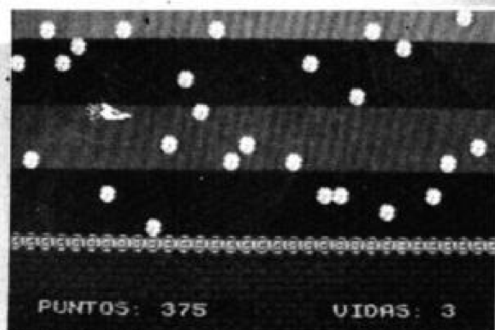
— Comprobar si chocamos, saltando a la línea 1300 en caso afirmativo.

Imprimir nuestra nave en las nuevas coordenadas. Se imprime también un asteroide y se realizan unos sonidos.

— Se incrementan los puntos y el contador del paisaje. Si éste último es mayor que 200 se pasa a la FASE 3.

— A continuación se llama a la rutina en C/M que hace el SCROLL y se repite el bucle.

En las líneas 1280-1330 está la rutina que hace estallar



nuestra nave, en caso de choque, y que nos resta una vida.

1350-1790. FASE 3. Esta es la última fase. Aquí hemos de recoger varios diskettes para completar nuestro objetivo. Las partes de que consta son:

1350-1380. Se informa de la fase 3.

1390-1530. Dibujo del escenario. Se dibujan las paredes, luego las puertas que las comunican y por último se sitúan todos los objetos que habrá en pantalla usando una subrutina que los imprime al azar en la misma, asegurán-

```
1250 LET A=USR 60100
1260 GO TO 1160
1270 REM
1280 REM * EXPLOSION *
1290 REM
1300 PRINT AT NX,5; FLASH 1;"iii
";: FOR A=1 TO 50: BEEP .01,RND*
30: NEXT A
1310 LET V=V-1: GO SUB 2070
1320 IF V<1 THEN GO TO 2230
1330 GO TO 1070
1340 REM
1350 REM * FASE 3 *
1360 REM
1370 PAPER 0: LET F=3: GO SUB 21
30
1380 REM
1390 REM * ESCENARIO *
1400 REM
1410 PRINT INK 2;AT 0,0;"aaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaa"
1420 LET CD=0: GO SUB 2070
1430 FOR A=1 TO 18
1440 FOR B=0 TO 31 STEP 5: PRINT
 INK 2;AT A,B;"a": NEXT B: PRINT
 INK 2;AT A,31;"a";
1450 NEXT A
1460 FOR A=5 TO 30 STEP 5: FOR B
=1 TO 3: PRINT AT 1+INT (RND*16)
,A;" ": NEXT B: NEXT A
1470 LET A$="g": INK 5
1480 FOR A=1 TO 20: GO SUB 2170:
 NEXT A
1490 LET A$="d": INK 4
1500 FOR A=1 TO 10: GO SUB 2170:
 NEXT A
1510 LET A$="h": INK 6
1520 FOR A=1 TO 5: GO SUB 2170:
NEXT A
1530 REM
1540 REM * INICIALIZA *
1550 REM
1560 INK 7: LET A$="f": GO SUB 2
170: LET XM=X: LET YM=Y
1570 LET A$="e": GO SUB 2170
1580 LET XB=X: LET YB=Y
1590 REM
1600 REM * BUCLE PRINCIPAL *
1610 REM
1620 LET MX=XM: LET MY=YM
1630 LET XM=XM+(1 AND XM<X)-(1 A
ND XM>X)
1640 LET YM=YM+(1 AND YM<Y)-(1 A
ND YM>Y)
1650 IF SCREEN$ (XM,YM)<>" " THE
N LET YM=MY
1660 IF SCREEN$ (XM,YM)<>" " THE
N LET XM=MX
1670 IF INKEY$="Q" THEN LET X=X-
1
1680 IF INKEY$="O" THEN LET Y=Y-
1
1690 IF INKEY$="P" THEN LET Y=Y+
1
1700 IF INKEY$="A" THEN LET X=X+
1
1710 PRINT AT XB,YB;" ";AT MX,MY
;" ";AT XM,YM;"f"
1720 LET A$=SCREEN$ (X,Y)
1730 IF A$="a" THEN LET X=XB: LE
T Y=YB
1740 IF A$="d" OR A$="f" OR A$="
g" THEN FOR A=7 TO 0 STEP -1: BE
EP .01,A: PRINT INK A;AT X,Y;"i"
: PAUSE 2: NEXT A: GO TO 1820
1750 IF A$="h" THEN FOR A=0 TO 6
```

```
0 STEP 2: BEEP .003,A: NEXT A: L
ET P=P+275: GO SUB 2080: LET CD=
CD+1: IF CD=5 THEN GO TO 1840
1760 PRINT AT X,Y;"e"
1770 LET XB=X: LET YB=Y
1780 GO TO 1620
1790 REM
1800 REM * PIERDE VIDA *
1810 REM
1820 LET V=V-1: IF V<1 THEN GO T
O 2230
1830 GO SUB 2080: LET A$="e": GO
 SUB 2170: GO TO 1760
1840 REM
1850 REM * CONSEGUIDO *
1860 REM
1870 CLS
1880 PRINT AT 4,0
1890 LET A$="!!! LO CONSEGUISTE
!!!": LET P=P+5000: GO SUB 2030
1900 PRINT : PRINT : LET A$="PUN
TOS:"+STR$ P: GO SUB 2030
1910 LET A$="RECORD:"+STR$ R
1920 IF P>R THEN LET R=P: LET A$
="!!! NUEVO RECORD !!!"
1930 GO SUB 2030
1940 PRINT : LET A$="OTRA PARTID
A: (S/N)": GO SUB 2030
1950 RANDOMIZE R
1960 POKE 60200,PEEK 23670: POKE
 60201,PEEK 23671
1970 IF INKEY$="S" THEN RUN 270
1980 IF INKEY$="N" THEN RANDOMIZ
E USR 0
1990 GO TO 1970
2000 REM
2010 REM * CENTRA TEXTO *
2020 REM
2030 PRINT INK 2+RND*5;TAB (16-L
EN A$/2);A$: PRINT : PRINT : RET
URN
2040 REM
2050 REM * MARCADORES *
2060 REM
2070 PRINT INK 2;AT 19,0;"aaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaa
            aaaa           aaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaa"
2080 PRINT INK 6;AT 20,2;"PUNTOS
: ";P;AT 20,21;"VIDAS: ";V
2090 RETURN
2100 REM
2110 REM * INDICADOR FASE *
2120 REM
2130 CLS : FOR A=1 TO [illegible]: FOR B=1
 TO 7: PRINT AT 11,12; INK B;"FA
SE ";F: BEEP .005,B: BEEP .005,A
: NEXT B: PAUSE A: NEXT A: CLS :
 RETURN
2140 REM
2150 REM * IMPRIME EN X,Y *
2160 REM
2170 LET X=1+INT (RND*17): LET Y
=1+INT (RND*29)
2180 IF SCREEN$ (X,Y)<>" " THEN
GO TO 2170
2190 PRINT AT X,Y;A$: BEEP .001,
25: RETURN
2200 REM
2210 REM * DESTRUIDO *
2220 REM
2230 BEEP 1,0: BEEP 1,-10: CLS :
 PRINT AT 4,0
2240 LET A$="GAME OVER": GO SUB
2030
2250 GO TO 1900
```

dose de que no hay ninguno debajo.

1540-1590. Se inicializan las coordenadas del vigilante y las de nuestro personaje.

1600-1790. Bucle principal de la fase 3. Aquí aparecen los siguientes pasos:

— Movimiento del vigilante, siempre en dirección hacia nosotros. Comprobando si choca contra las paredes.

— Luego se lee el teclado.

— Se imprime el malo.

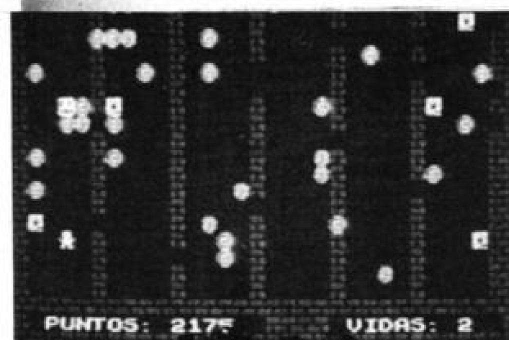
— Se obtiene el carácter sobre el que se situará nuestro protagonista, y se comprueba si se trata de un muro, ladrillo, roca o diskette.

— Ahora se imprime nuestro muñeco, repitiéndose el bucle.

1800-1830. Rutina de pérdida de vida de la FASE 3.

1840-1990. Rutina de conseguido. Se comprueba si hemos logrado record y se aguarda la decisión de volver a jugar o de abandonar el juego.

A partir de la línea 2000 están las subrutinas que usa el programa de un modo general. Estas son:



2000-2040. Rutina que imprime el texto centrado.

2050-2090. Imprime marcadores de la parte inferior.

2100-2140. Subrutina indicadora de FASE.

2150-2200. Imprime al azar el carácter contenido en A$ en un lugar vacío de la pantalla.

2210-2250. Rutina de final. Mensaje GAME OVER y salto a la línea 1900.

DYZZY

COMENTARIO

DYZZY es un juego divertido cuyo protagonista es un simpático "huevo" que trata de destruir al mago ZAKS haciendo una pócima mágica en un caldero al que habrá de arrojar todos los ingredientes que vaya recogiendo por las pantallas del juego.

Los gráficos de éste juego son muy variados y coloristas y los movimientos de DYZZY y del resto de figuras son excelentes. La presentación del juego está muy cuidada e incorpora una excelente melodía. Todos los escenarios han sido cuidados y ofrecen multitud de detalles simpáticos, muchos de ellos difíciles de programar.

La aventura en si no es excesivamente original pero la trama del juego es entretenida y averiguar los diferentes trucos para pasar cada una de las dificultades que se nos presentan llega a convertirse en una cuestión de "honrilla", por lo que es difícil separarse del SPECTRUM sin haber llegado un poco más lejos que la vez anterior.

DYZZY estaba explorando el bosque encantado en busca de flores, bayas y un palo con el que hacerse un bastón, cuando descubrió una misteriosa lápida. Limpiando el lodo que la cubría le fue posible leer la siguiente inscripción:

La Pócima Avawiffover

DYZZY recordó que su padre le había hablado de aquella pócima "es la única manera de liberar a nuestro país destruyendo al malvado brujo ZAKS". DYZZY TEMBLO ASUSTADO cuando recordó éstas palabras.

ZAKS trajo el miedo al pueblo, con sus hechizos que hacen que la gente se envejezca, vuelve ciegos a los hombres y hace que llueva los domingos por la tarde a la hora del CRICKET. DYZZY estaba decidido a poner fin a todo esto y convertirse en el heroe del lugar.

ORIGINALIDAD GRAFICOS MOVIMIENTOS SONIDO DIFICULTAD ADICCION PRESENTACION MEDIA

DRUID

DRO-SOFT

Hasta ahora se ha mantenido el equilibrio político y ha habido paz por todo Berlon. Pero a través de una puerta inter-dimensional en las mazmorras del maléfico nigromante Acamantor, han aparecido ahora cuatro principes endemoniados.
Ha recaido sobre ti, el último de los Grandes Druidas, la tarea de destruir a los principes y de cerrar la puerta.
Los principes endemoniados andan acechando por las tenebrosas profundidades de la mazmorra, y solamente pueden destruirse por medio de tu poderoso hechizo, que en este caso debe de transportarse por medio del tacto.

Por toda la mazmorra encontrarás arcas que contienen hechizos de una fuerza formidable para ayudarte en tu búsqueda y Pentogramas de vida, que revitalizarán totalmente tu esencia de vida. Usa estos recursos con prudencia a fin de sobrevivir el ataque constante de los monstruos dimensionales infernales y poder hacer frente a los principes endemoniados.

Triunfa y alcanzarás el alto puesto de Maestro de la luz, el mayor de todos los Druidas. Fracasa y no serás más que un mediocre ingenio y tu alma cautiva se reunirá con los numerosos fantasmas de los que te han precedido, perdidas en las mazmorras de Acamantor.

DRUID es un juego en el que tendrás que recorrer un intrincado laberinto buscando los hechizos que te ayuden a destruir a las fuerzas del mal.
La presentación es muy espectacular, ya que junto a un buen dibujo de portada, al mismo tiempo que al efectuar la carga del programa, podemos escuchar una melodía muy bien adaptada para el ordenador.
DRUID tiene una gran abundancia de gráficos muy elaborados, sin embargo el espacio "útil" donde se desarrolla el juego es algo reducido. En él, el laberinto se desplaza mediante un "SCROLL" en cuatro sentidos muy suave.
Los sonidos de ambiente, los movimientos y la rapidez de respuesta son otra buena característica de DRUID.
La dificultad no es muy elevada pero hay que estar muy atento a los múltiples enemigos que van apareciendo en pantalla a lo largo de la aventura.

FUNKY PUNKY



FUNKY PUNKY es una aventura en la que su protagonista sale de su habitación para acudir a un examen. Para ello deberá recoger una serie de objetos que hay dispersos por las diferentes habitaciones de la casa y que le son necesarios para coger su moto y poder acudir al instituto, sorteando el tráfico, para realizar su examen.

Podemos dividir éste juego en cuatro partes: la primera transcurre dentro de la casa y en ella podemos contemplar una gran variedad de gráficos muy vistosos y elaborados, el nivel de dificultad de ésta fase no es muy elevado y basta ir encontrando los diferentes objetos que aumentan nuestra energía y esquivando a los enemigos para pasar por el laberinto de pantallas que forman la casa. Una vez fuera de ella y con las llaves de la moto en la mano, se pasa a la segunda fase donde hay que ir adelantando los coches que vamos alcanzando, ésta etapa es bastante difícil de superar y en ella podemos apreciar un desplazamiento o scroll rápido y suave, y el movimiento de la moto también muy conseguido.

La tercera fase comienza al llegar al instituto; otro laberinto de pantallas similar al de la casa, donde hay que econtrar el aula del examen. Una vez en ella, y si llevamos los objetos pertinentes, comienza la cuarta y última fase: el examen, que consta de 10 difíciles preguntas a las que hay que responder afirmativa o negativamente.

En general los gráficos están muy elaborados aunque no hay una gran variedad de enemigos.

La pantalla del juego presenta un simpático dibujo acompañado de una melodía alegre y pegadiza.

La idea de la aventura es muy original aunque utilice el clásico recurso de las pantallas en forma de laberinto.

Como hemos dicho, la melodía inicial es muy buena y durante el resto del juego algunos sonidos van ambientando ligeramente las distintas situaciones.

¡Ahrrrg!, ¡qué dolor de cabeza!. No vuelvo a dar otra fiesta hasta que no pasen dos meses. ¡Tomá! si hoy tengo un examen de evaluación a las 10; y todavía aquí...

Realmente FUNKY PUNKY lo tiene difícil. Ha de ir al instituto para realizar el examen que tiene a las 10 de la mañana. Son las 8 y todavía tiene que recoger todas sus cosas revueltas entre los desperdicios de la fiesta que tuvo lugar en su apartamento la noche anterior.

Cuando las reuna, ha de coger su ciclomotor e ir al instituto atravesando las peligrosas calles de la ciudad, llenas de conductores medio dormidos que se dirigen a su trabajo.

Una vez allí, deberá encontrar el aula donde se va a realizar el examen y afrontar con toda resignación las diez preguntas que decidirán su aptitud para superar el curso.

Para realizar el examen será necesario que lleve su pluma y la calculadora científica.

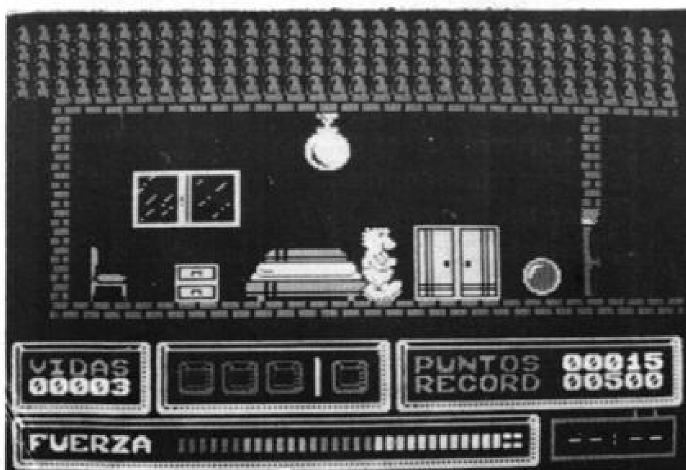
Por supuesto, no ha desayunado y para mantener sus energías a un alto nivel deberá tomar los DONUTS y beber los botellines que encuentre. Un cigarrillo después de la comida tampoco le sentará mal.

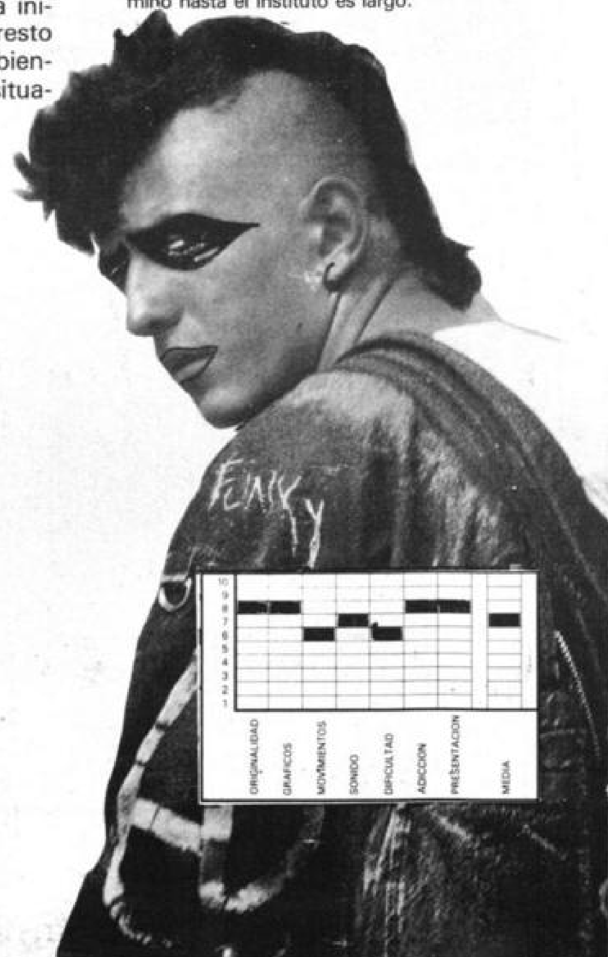
Los efectos de la resaca harán que vea bichos extraños que tendrá que evitar si desea continuar sus andanzas.

Para ver el tiempo que le resta hasta la hora del examen, deberá recoger el reloj. Y para poder abrir el candado de la moto tendrá que usar su llave.

¡Deprisa!, queda poco tiempo y el camino hasta el instituto es largo.







THE SENTINEL

El objeto del juego es absorber al "Sentinel" y sustituirlo como regidor del panorama. Una vez logrado, el jugador puede suprespaciar a un nuevo mundo y tratar de lograrlo de nuevo.

Una vez que se haya insertado el cassette con el juego, aparecerá el título en la pantalla. Al pulsar cualquier tecla se le pedirá que inscriba un número de panorama, del 0000 a 9999.

Una vez inscrito el número de panorama y pulsar retorno, se le pedirá que marque un número de clave de 8 cifras, al menos que haya elegido panorama 0000, que no lo requiere.

Pasados unos segundos, aparecerá la superficie panorámica. El Sentinel y los Sentries permanecerán inactivos hasta que el jugador gaste o absorba energía. Esto le permitirá mirar alrededor y plantear su ataque al Sentinel.

Una vez activados, el Sentinel y los Sentries giran despacio, barriendo el panorama, buscando cuadros que contienen más de 1 unidad de energía. Si pueden ver con claridad uno de estos cuadros, el Sentinel/Sentry reducirá la engergiá a 1 unidad, absorbiendo 1 unidad a la vez y creando un árbol al azar en el panorama. De esta forma, el robot se convierte en bloque y el bloque se convierte en árbol.

Se absorbe y se crean cosas, haciendo girar las miras y centrándolas en la superficie del cuadro bajo el objeto a ser absorbido/creado. Sin embargo los bloques actúan como extensión de la superficie del cuadro, y las miras deberan ser dirigidas al costado del bloque, se pueden apilar los bloques y colocarse cosas sobre ellos.

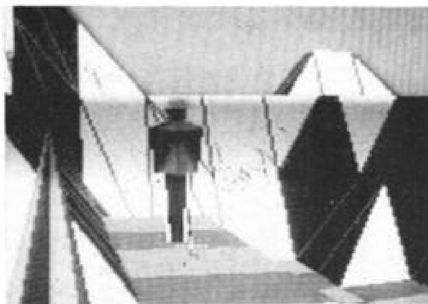
La energía disponible aparece mostrada en la parte superior izquierda de la pantalla en forma de iconos de robot, bloque y árbol. Un robot de color (no azul) vale por 15 unidades de energía.

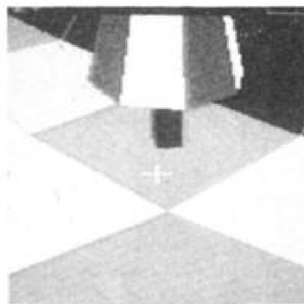
Para moverse alrededor del panorama, se debe crear un robot y entonces, con las mairas aún encendidas, púlsese la tecla de transferencia. Así se pasará al nuevo robot, de cara al antiguo robot, que puede ahora ser absorbido. El indicador en la parte superior derecha de la pantalla es el de advertencia de barrido. Si le localiza un Sentinel/Sentry, el indicador aparecerá lleno de motas. Tiene aproximadamente 5 segundos para desaparecer de vista antes de que se le agote la energía, 1 punto a la vez. Una vez agotadas toda la energía, queda absorbido totalmente y acaba la jugada.

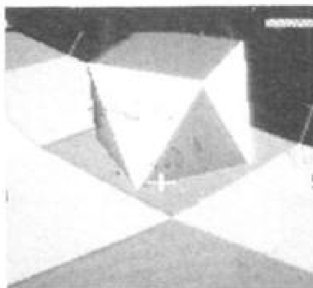
El volumen de energía total en el panorama permanece constante, de forma que si se absorbe 1 unidad de su energía, se colocará al azar un árbol en el panorama.

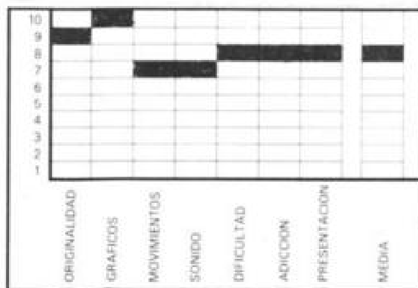
Si el indicador de advertencia de barrido sólo se encuentra medio lleno de motas, esto significará que el Sentinel/Sentry puede verle pero no puede ver el cuadro en el que permanece usted situado.











SENTINEL

SENTINEL es un juego sobrecogedor tanto por su temática como por los desoladores escenarios en los que transcurre la acción.

Podríamos calificar a SENTINEL como un simulador de una realidad futura: el control de un robot que nos trasmite la visión de otro mundo a través de un monitor de vídeo.

Los ESCENARIOS (10.000 en total) representa diferentes caras de un PLANETA cuyo paisaje es un enjambre de figuras geométricas.

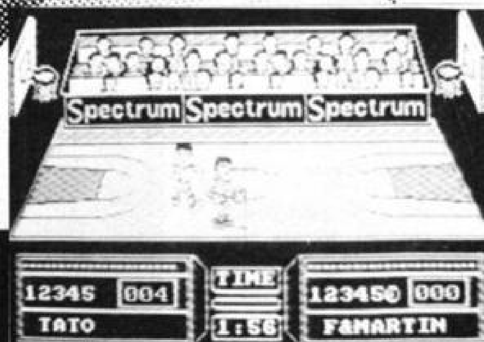
Hemos de descubrirnos ante el método de elaboración de los gráficos, así como también es de destacar la originalidad del tema y su ejecución.

La pantalla de presentación y los sonidos que amenizan y dan ambiente al juego también tienen una gran calidad.

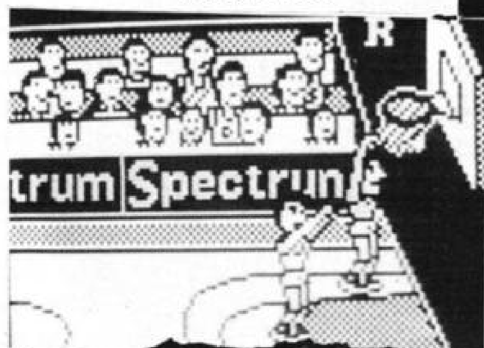
SENTINEL es un juego que puede parecer en un principio complicado de manejar, no sólo por el número de teclas que utiliza sino por la cantidad de situaciones y posibilidades que ofrece; pero una vez conocido su manejo es apasionante.

Los movimientos son algo lentos, pero esto es totalmente disculpable dada la complejidad que entraña mover en pantalla unos paisajes tan elaborados.

FERNANDO MARTIN
Basket Master





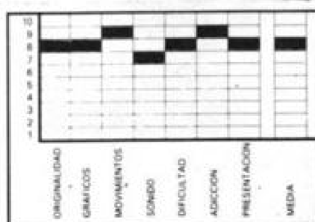


Los juegos de "simulación" entrañan a la hora de programarlos una dificultad superior a los demás dado que necesitan ajustarse lo más posible a la realidad que simulan, no basta con que el juego funcione bien y sea divertido sino que además debe parecer real, y esto implica un duro trabajo.

Fernando Martín Basket Master cumple perfectamente todo lo exigible a un juego de simulación, todos sus detalles están cuidadosamente estudiados para conseguir un juego excitante y divertido por su "realismo".

En la pantalla de presentación aparece una simpática caricatura del famoso jugador ligeramente animada y con una buena melodía. El resto de sonidos del juego están destinados a proporcionarle unos adecuados efectos ambientales. Los gráficos como ya hemos mencionado están muy elaborados y consideran múltiples posturas de los jugadores así como la agitación del público en cada canasta.

El nivel de dificultad puede ser seleccionado por el jugador pudiendo enfrentarse a un segundo jugador o bien al ordenador que hace las veces de Fernando Martín.



HISTORIA

Cuando salgas a la cancha te encontrarás nervioso, preocupado, con deseos de hacer un buen partido y con miedo de fallar los tiros de tres puntos.

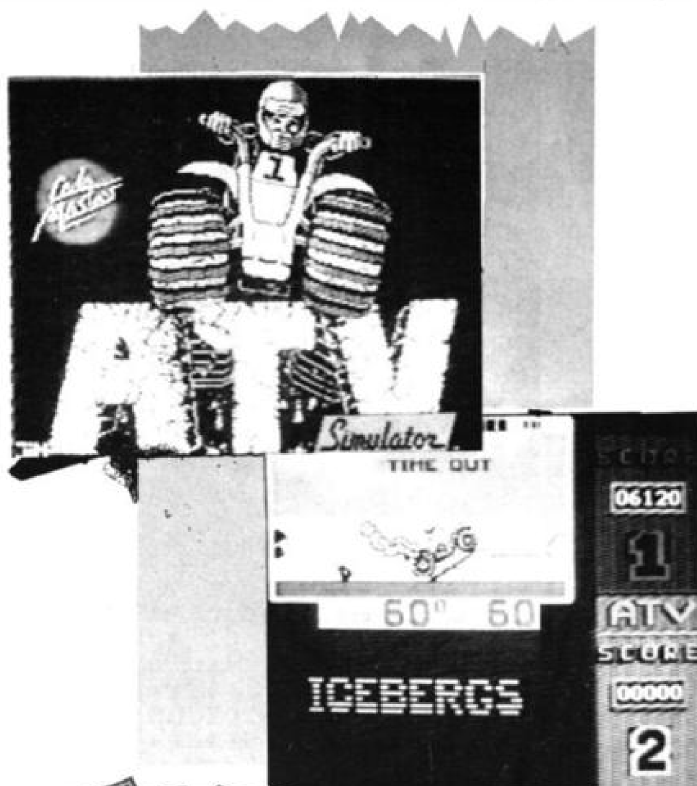
Imagina que, además, tienes que enfrentarte a Fernando Martín, a un auténtico número uno, que no va a perdonarte.

Todos tus músculos se ponen en tensión, la adrenalina fluye por tus venas y tu cerebro debe averiguar cuáles son los puntos débiles de tu adversario.

El baloncesto es el deporte que mejor conjuga el desarrollo físico del ser humano, según un ordenador especializado en deportes del Istituto Tecnológico de Masachusets.

Fernando Martín Basket Master es una prueba clara de que también el baloncesto puede desarrollar la mente.

ALL TERRAIN VEHICLE SIMULATOR



Completa cada recorrido en el plazo de tiempo disponible sin consumir todo el combustible. Usa los mandos de izquierda/derecha para dirigirte hasta la ATV y luego salta a él mirando en el mismo sentido y comienza el recorrido. Usa arriba/abajo para acelerar/frenar. Salta para lograr remonte adicional cuando subas por una rampa y también para superar orillas certadas sin chocar. Para ascender por pendientes pronunciadas, wheelie lentamente hacia ellas y salta repetidamente para subir.

Si chocas y sales despedido, salta para levantarte y luego montarte en tu ATV.



ATV (ALL TERRAIN VEHICLE)

"Vehículo todo terreno" es un juego de simulación en el que debes ir superando cada una de sus distintas etapas sin rebasar el tiempo máximo permitido, el reto es emocionante y una vez se empieza a jugar cuesta trabajo dejarlo.

La pantalla de portada presenta un buen dibujo durante la carga del programa a la que sigue una buena melodía muy bien adaptada al SPECTRUM.



Los gráficos son buenos y variados y los movimientos del conductor y su vehículo son suaves y ágiles, sin embargo la superficie "útil" en pantalla, es decir, la ventana en la que se desarrolla el juego es muy reducida.



ATV presenta también la posibilidad de establecer una competición con un segundo jugador.

Los sonidos son adecuados y cada vez que se supera una etapa cambia el paisaje y el nombre de la pantalla se imprime en grandes caracteres.

ORIGINALIDAD · GRÁFICOS · MOVIMIENTOS · SONIDO · DIFICULTAD · ADICIÓN · PRESENTACIÓN · MEDIA

MUROS GALACTICOS

Como controlador espacial de la frontera electromecánica entre los dos polos de atracción de los planetas XENOS y COXENOS, has de impedir que las naves espaciales y los satélites móviles choquen contra éstas barreras.

Para impedir que colisionen puedes abrir un hueco en cada barrera empleando las teclas:

Q. Abre barrera de la izquierda.

P. Abre barrera de la derecha.

COMENTARIO DEL PROGRAMA

En éste programa el elemento fundamental para el éxito es sincronizar bien los movimientos de la nave y el satélite con el de apertura de las barreras.

10-220. Inicialización y presentación. Se fija cursor en minúsculas. Se redefinen los UDGs. y se fijan los colores POKEando directamente en las variables del sistema. Luego se imprime el títuio y tras una pausa comienza el juego.

230-350. Dibuja el paisaje. Fija los colores. Imprime los marcadores y las estrellas.

360-410. Dibujo de los muros. Se realizan las dos barreras.

420-550. Se inicializan las variables que se van a usar durante el juego.

560-860. Bucle principal del juego. Se lee el teclado. Se imprimen las barreras y a continuación lo dos objetos móviles. Las comprobaciones son hechas por medio del color, con la función ATTR.

870-960. Rutina de choque del OVNI con la barrera.

970-1020. Rutina de choque del planeta con la barrera.

1030-1060. Subrutina que incrementa e imprime los puntos.

1070-1250. Final por fallos. Se imprimen los

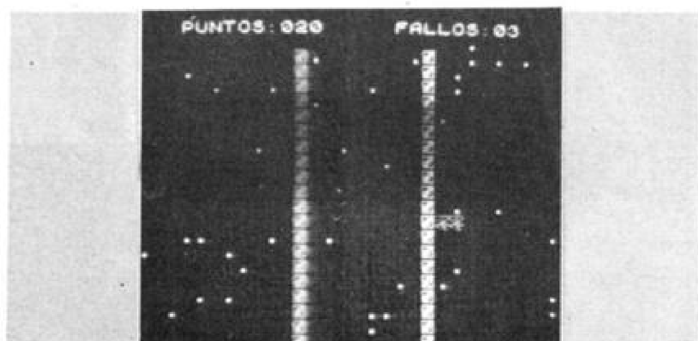


mensajes correspondientes, se comprueba si batimos el record y se toca una corta meiodía.

1260-1340. Melodía de R.I.P.

1350-1460. Esta es la subrutina más curiosa del programa, imprime una frase en el interior de un recuadro (ajustado por supuesto al tamaño de la frase con un color al azar.

1470-1610. Redefinición de los U.D.G.s. mediante un bucle, se leen las sentencias datas de las líneas 1560-1610, y se almacena el valor leído en la zona de los U.D.G.s.




```
  10 REM ********************
  20 REM * MUROS GALACTICOS *
  30 REM ********************
  40 REM
  50 REM - FIJA MINUSCULAS -
  60 REM
  70 POKE 23658,0
  80 REM - REDEFINE UGDS -
  90 GO SUB 1500
 100 REM
 110 REM - FIJA COLORES -
 120 REM
 130 POKE 23693,7
 140 BORDER 0
 150 CLS
 160 REM
 170 REM - TITULO -
 180 REM
 190 PRINT AT 9,0
 200 LET A$="MUROS GALACTICOS"
 210 GO SUB 1380
 220 PAUSE 200
 230 REM
 240 REM - DIBUJA PAISAJE -
 250 REM
 260 BORDER 0
 270 PAPER 0
 280 INK 7
 290 CLS
 300 PRINT AT 0,0;"    PUNTOS:00
0      FALLOS:00     "
 310 REM - IMPRIME ESTRELLAS -
 320 FOR a=1 TO 50
 330 PRINT AT 1+INT (RND*20),INT
 (RND*31);"."
 340 NEXT a
 350 REM
 360 REM - DIBUJA MUROS -
 370 REM
 380 FOR a=2 TO 21
 390 PRINT AT a,12; INK 3;"A";AT
 a,21; INK 6;"A"
 400 NEXT a
 410 REM
 420 REM -INICIALIZA VARIABLES-
 430 REM
 440 LET X1=10
 450 LET D1=0
 460 LET X2=10
 470 LET D2=0
 480 LET y=0
 490 LET yb=y
 500 LET g=500
```

```
 510 LET z=30
 520 LET zb=z
 530 LET vi=0
 540 LET pu=0
 550 REM
 560 REM - BUCLE PRINCIPAL -
 570 REM
 580 IF D1=0 THEN IF INKEY$="q"
THEN LET D1=-1
 590 IF D1=-1 THEN PRINT AT 2+X1
,12;" ";AT 21-X1,12;" "
 600 IF D1=1 THEN PRINT AT 2+X1,
12; INK 3;"A";AT 21-X1,12; INK 3
;"A"
 610 LET X1=X1+D1
 620 IF X1<1 THEN LET D1=1
 630 IF X1>9 THEN LET D1=0
 640 IF D2=0 THEN IF INKEY$="p"
THEN LET D2=-1
 650 IF D2=-1 THEN PRINT AT 2+X2
,21;" ";AT 21-X2,21;" "
 660 IF D2=1 THEN PRINT AT 2+X2,
21; INK 6;"A";AT 21-X2,21;"A"
 670 LET X2=X2+D2
 680 IF X2<1 THEN LET D2=1
 690 IF X2>9 THEN LET D2=0
 700 PRINT AT 10,yb;"  ";AT 11,y
b;"  "
 710 PRINT INK 2;AT 10,y;"FG";AT
 11,y;"HI"
 720 LET yb=y
 730 LET y=y+1
 740 IF y>29 THEN LET y=0: GO SU
B 1030
 750 IF y=15 THEN LET g=1000
 760 IF ATTR (10,y+1)=3 OR ATTR
(10,y+1)=6 THEN GO SUB 980
 770 IF g=1000 THEN GO SUB 800
 780 IF g=500 THEN GO TO 580
 790 GO TO 580
 800 PRINT AT 13,zb;"  ";AT 13,z
; INK 4;"CD"
 810 LET zb=z
 820 LET z=z-1
 830 IF z<0 THEN LET z=30: GO SU
B 1030
 840 IF ATTR (13,z)=3 OR ATTR (1
3,z)=6 THEN GO TO 890
 850 RETURN
 860 REM
 870 REM - OVNI CHOCA -
 880 REM
 890 PRINT AT 13,z+1; INK 4;"EE"
 900 BEEP .2,-10
 910 PRINT AT 13,z+1;"
 920 LET z=30
 930 LET vi=vi+1
 940 PRINT AT 0,27;vi
 950 IF vi=8 THEN GO TO 1100
 960 GO TO 850
 970 REM - PLANETA CHOCA -
 980 PRINT AT 10,y-1; INK 2;"EE"
;AT 11,y-1;"EE"
 990 BEEP .2,-10
1000 PRINT AT 10,y-1;"  ";AT 11,
y-1;"  "
1010 LET y=0
1020 GO TO 930
1030 LET pu=pu+10
1040 PRINT AT 0,12;pu
1050 BEEP .05,10
1060 RETURN
1070 REM
1080 REM - FINAL POR FALLOS -
1090 REM
1100 LET p=pu
1110 BORDER 0
1120 POKE 23693,7
1130 CLS
1140 PRINT AT 4,0
1150 LET a$="GAME OVER"
1160 GO SUB 1380
1170 LET A$="TU PUNTUACION: "+ST
R$ (P)
1180 GO SUB 1380
1190 GO SUB 1280
1200 LET A$="OTRA PARTIDA: (S/N)
"

1210 GO SUB 1380
1220 IF INKEY$="S" OR INKEY$="s"
 THEN GO TO 260
1230 IF INKEY$="n" OR INKEY$="N"
 THEN RANDOMIZE USR 0
1240 GO TO 1220
1250 REM
1260 REM - MELODIA -
1270 REM
1280 DATA .5,-3,.55,-3,.2,-3,.65
,-3,.5,0,.25,-1,.5,-1,.25,-3,.5,
-3,.25,-4,.75,-3
1290 RESTORE 1280
1300 FOR a=1 TO 11
1310 READ aa,aaa
1320 BEEP aa,aaa
1330 NEXT a
1340 RETURN
1350 REM
1360 REM - IMPRIMIR A$ -
1370 REM
1380 LET b$="██████████████████
█████████████"
1390 INK 2+RND*5
1400 LET c=LEN a$+2
1410 LET b=16-c/2
1420 PRINT TAB (b);b$( TO c)
1430 PRINT TAB (B);"█"; INVERSE
1; PAPER 9;a$; INVERSE 0;"█"
1440 PRINT TAB (b);b$( TO c)
1450 PRINT
1460 RETURN
1470 REM
1480 REM - DATAS U.D.G. -
1490 REM
1500 RESTORE 1560
1510 FOR A=0 TO 71
1520 READ B
1530 POKE USR "A"+A,B
1540 NEXT A
1550 RETURN
1560 DATA 254,158,190,250,250,24
2,254,0,0
1570 DATA 255,153,231,153,231,15
3,255,57
1580 DATA 15,62,255,117,63,28,48
,156,240,124
1590 DATA 255,174,252,56,12,170,
17,164,24
1600 DATA 106,149,72,20,3,14,61,
53,123,91,238,255,192,240,188,22
0,126,174
1610 DATA 223,243,255,249,107,11
7,61,59,15,3,247,175,110,238,220
,188,240,192
```

A B C D E F G H I

FICHERO

```
BUSQUEDA DE FICHA..............

FICHA: 2
NOMBRE....:PEPITO
DIRECCION.:C/ PERCEBE
TELEFONO..:4444444
C. POSTAL.:11111
POBLACION.:LEP
```

```
EN LA MEMORIA CABEN 659 FICHAS
           FICHA:2
NOMBRE....:JUANITO
DIRECCION.:C/ ANCHOA
TELEFONO..:55555
C. POSTAL.:22222
POBLACION.:LEPE TAMBI

DATOS CORRECTOS ?..............
```

Este puede ser un programa muy útil para aquellos que poseen gran cantidad de cosas susceptibles de ser catalogadas.

Como su propio nombre indica su función es la de servir de fichero flexible-multiuso. ¿Qué es eso de "flexible-multiuso"? Pues sencillamente que si lo que a uno le interesa es disponer de una especie de agenda de direcciones y teléfonos, puede definir el formato de las fichas más o menos así:

Nombre ... (ejemplo 1)
Dirección ....
Tlf. ....

Si por el contrario, lo que necesita es un archivo de notas, sólo tiene que definir las fichas como:

Curso: .... (ejemplo 2)
Trimestre: ....
Física: ....
Química: ....
Etc. ....

Es decir, en este programa se pueden definir fichas con campos y el nombre que se desee, de la longitud que se desee y en la longitud que queramos (con el único tope de la memoria del ordenador, claro está).

Cuando se pone en marcha el programa hay que esperar durante unos instantes a que éste cargue en memoria el bloque de código máquina. Seguidamente preguntará por la cantidad de términos o campos de que constará cada ficha. (Si siguiéramos el ejemplo 1 responderíamos con el número 3). Después irá preguntando por el nombre de cada término, que no podrá exceder de 10 caracteres, y por la longitud máxima que va a tener (no hay limitación, salvo la que imponga la memoria).

Una vez cumplimentados estos requisitos obligatorios aparece en pantalla el menú principal, que consta de 9 opciones:

1.— INTRODUCIR FICHAS:

Al pulsar esta opción, desaparece el menú y se indica en la parte superior de la pantalla el número de fichas que caben en la memoria, así como el número de ficha que se está introduciendo.

Cuando se termina de escribir una ficha el ordenador pregunta si los datos son correctos y en caso de responderse negativamente procedería a pedir la introducción de esa ficha nuevamente. Para salir de esta opción basta con pulsar el comando OUT (el comando que está en la tecla 0, no las letras O, U y T) a la pregunta del término de la siguiente ficha.

2.— CORREGIR FICHAS YA HECHAS:

Para usar esta opción hay que buscar previamente la ficha que se desea modificar con la opción 3, volver al menú y pulsar esta opción.

A la pregunta de "TERMINO A MODIFICAR?" hay que responder con el nombre del término que se desea corregir.

3.— BUSQUEDA DE FICHA:

Supongamos que queremos ver todas las fichas que hay en memoria. Entonces nos bastaría con responder con la letra S a la pregunta "PISTA?".

Si por el contrario, queremos buscar (en el caso de que hubiéramos elegido el formato de ficha del ejemplo 1) todas las fichas cuyo nombre empiece por ANT, entonces haríamos:

PISTA? NOM-

Para los curiosos, he aquí el listado ensamblador de las rutinas de borrado, búsqueda y ordenación alfabética de fichas, así como de la pequeña rutina que "engorda" las letras.

```
LD   HL,15616
LD   DE,64437
LD   BC,00768
LD   A,(HL)
SRA  A
OR   (HL)
LD   (DE),A
INC  DE
INC  HL
DEC  BC
LD   A,C
OR   B
JR   NZ,40969
LD   HL,64181
LD   (23606),HL
RET
```

BRE:ANT (ENTER)
y obtendríamos lo que deseamos. Si tenemos un fichero de libros y queremos encontrar todos los libros de, por ejemplo, Cervantes, escribiríamos:

PISTA? AUTOR: MIGUEL DE CERVANTES

En el caso de que lo que busquemos no exista, el ordenador responderá con "NO HAY MAS FICHAS" y retornará al MENU.

4.— BORRAR FICHAS:

Esta opción es como la 2. Primero buscamos con la opción 3 la ficha que queremos suprimir, retornamos al menú y pulsamos esta opción.

5.— ORDENAR ALFABETICAMENTE:

Esta opción ordena alfabéticamente (más correctamente, ordena en función de los códigos ASCII) las fichas en función del término que le indiquemos.

6.— GRABAR FICHERO:

Se pide un nombre (no mayor de 8 caracteres) y se procede a grabar y verificar el fichero en una cassette.

7.— CARGAR FICHERO:

Como la 6, sólo se carga un fichero de una cassette a memoria. ¡¡ATENCION!!, si se intenta cargar un fichero con distinto formato de fichas al definido al comenzar el programa los resultados pueden ser catastróficos.

8.— NUEVO TIPO DE FICHERO:

Permite redefinir un nuevo formato de fichas distinto al definido al comienzo del programa. En realidad esta opción hace una llamada al inicio del programa, por lo que el fichero que hubiera en memoria antes de usar esta opción quedaría inutilizado.

9.— IMPRESORA:

Como su nombre indica pasa a impresora las fichas comprendidas entre el número puesto en la pregunta "DESDE LA FICHA?" y el indicado en la pregunta "HASTA LA FICHA?" inclusive. Si en la primera pregunta se contesta con cero, el ordenador listará todas las fichas.

En cualquier momento el listado se puede interrumpir pulsando cualquier tecla, con lo que se retornaría al menú.

Si por cualquier motivo hubiera un retorno al BASIC, poniendo GO TO ME se regresaría al MENU sin pérdida de datos.

Ejemplo de cómo lista a impresora unas fichas dedicadas en este caso a libros y que fueron definidas del siguiente modo:

```
FICHA: 1
TITULO....DON QUIJOTE

AUTOR.....MIGUEL DE CERVANTES

EDITORIAL.PLANETA AGOSTINI
A/O.......1983
GENERO....NOVELA
--------------------------------

FICHA: 2
TITULO....HAMLET

AUTOR.....WILLIAM SHAKESPEARE

EDITORIAL.ANAGRAMA
A/O.......1986
GENERO....DRAMA
--------------------------------

FICHA: 3
TITULO....LA CELESTINA

AUTOR.....FERNANDO DE ROJAS

EDITORIAL.SEIX BARRAL
A/O.......1979
GENERO....NOVELA
--------------------------------

FICHA: 4
TITULO....LA COLMENA

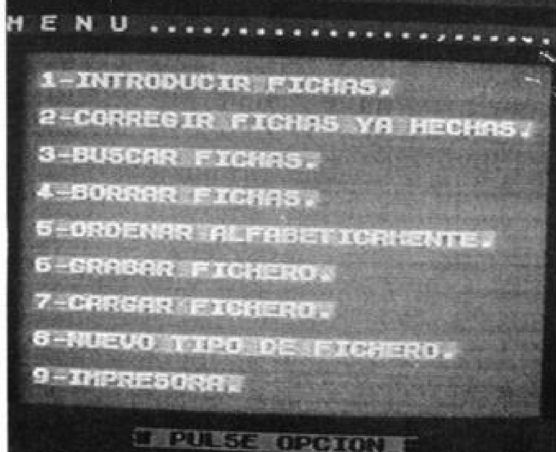
AUTOR.....CAMILO JOSE CELA

EDITORIAL.PARANINFO
A/O.......1981
GENERO....NOVELA
--------------------------------

FICHA: 5
TITULO....ROMANCERO GITANO

AUTOR.....ANTONIO MACHADO

EDITORIAL.LABOR
A/O.......1965
GENERO....POESIA
--------------------------------
```



PROGRAMA FICHERO

```
   1 REM FICHEROS#@A.M.G. SOFTWA
RE 1986
   2 CLEAR VAL "64436": BORDER P
I/PI: PAPER NOT PI: INK VAL "9":
 CLS : POKE VAL "23609",VAL "15"
: PRINT AT VAL "10",NOT PI;"ESPE
RE UN MOMENTO"
   3 FOR N=PI/PI TO INT (EXP 1):
 READ A$: FOR T=PI/PI TO LEN A$-
PI/PI STEP VAL "2": LET B=16*(CO
DE A$(T)-48-7*(A$(T)>"9"))+(CODE
```

```
 A$(T+1)-48-7*(A$(T+1)>"9")): PO
KE 23296+(41944 AND N>1)+INT (T/
2),B: NEXT T: NEXT N
   4 DATA "21003D11B5FB0100037EC
B2FB6121323087980020F421B5FA22365
CC9","2147AE11E3AD010100ED80C93A
BFFE673AC1FEBCF83ABEFE6F3AC2FE47
3AC3FE5F16001911 0E5B1ABE20052313
10F8C91164003ABEFE6F3ABFFE67197D
32BEFE7C32BFFE18C6002AC4FE22C6FE
2ABCFE22D0FE2ABAFE22D2FE2ACAFEED
4BC8FE03ED43C8FEB7ED42F801000003
2ACCFEB7ED42FA87FF2AD0FE09562AD2
FE097E8A28E9F287FF0101002AD0FEED
5BCEFEB7ED52097EED5BD4FE195677B7
7AED5BD4FEED527703EBB7ED42FA87FF
18DA2AC6FEED5BD4FE1922C6FE2AD0FE
1922D0FE2AD2FE1922D2FEED5BD6FE2A
C6FE2387ED52FA41FFC31FFF"
   5 RANDOMIZE USR VAL "23296":
LET ME=VAL "100": LET Z=PI/PI: L
ET NFI=Z: LET BY=VAL "256": POKE
 VAL "23658",VAL "8"
   6 LET P1=VAL "65216": LET P2=
VAL "65217": LET D1=VAL "65214":
 LET D2=VAL "65215"
   7 LET RO=VAL "9998": DEF FN A
(X)=INT (X/BY): DEF FN B(X,Y)=X-
BY*PEEK Y: DEF FN C(X)=PEEK X+BY
*PEEK (X+Z)
  30 INPUT "CUANTOS TERMINOS MAN
EJARA ? ";T: DIM T$(T,VAL "10"):
 LET NC=NOT PI: DIM H(T)
  40 FOR I=Z TO T: INPUT "NOMBRE
 DEL TERMINO ";(I);":"; LINE E$:
 FOR H=Z TO VAL "10"-LEN E$: LET
 E$=E$+".": NEXT H: LET T$(I)=E$
  60 INPUT "NO. DE CARACTERES DE
L TERMINO ";(I);":";H(I): LET NC
=NC+H(I): NEXT I
  70 LET L=FN C(VAL "23641")-FN
C(VAL "23635")+VAL "600": LET NF
=INT ((VAL "64436"-FN C(VAL "236
35")-NC-L-VAL "150")/NC): LET DI
=L+FN C(VAL "23635"): LET DIR=DI
+NC
  80 LET PUN=DIR
  85 LET R$=".........."
  90 REM MENU DE OPCIONES
 100 CLS : LET C$="M E N U .....
....................:": GO SUB RO
 110 FOR Y=VAL "3" TO VAL "21":
PRINT AT Y,Z; PAPER VAL "2";"
                            ": NE
XT Y
 120 PRINT AT VAL "4",VAL "2"; P
APER VAL "3";"1-INTRODUCIR FICHA
S.";AT VAL "6",VAL "2"; PAPER VA
L "3";"2-CORREGIR FICHAS YA HECH
AS.";AT VAL "8",VAL "2"; PAPER V
AL "3";"3-BUSCAR FICHAS.";AT VAL
 "10",VAL "2"; PAPER VAL "3";"4-
BORRAR FICHAS.";AT VAL "12",VAL
"2"; PAPER VAL "3";"5-ORDENAR AL
FABETICAMENTE.";AT VAL "14",VAL
"2"; PAPER VAL "3";"6-GRABAR FIC
HERO.";AT VAL "16",VAL "2"; PAPE
R VAL "3";"7-CARGAR FICHERO.";AT
 VAL "18",VAL "2"; PAPER VAL "3"
;"8-NUEVO TIPO DE FICHERO.";AT V
AL "20",VAL "2"; PAPER VAL "3";"
9-IMPRESORA."
 130 PRINT #NOT PI;AT Z,VAL "8";
 INK NOT PI; PAPER VAL "5";"# PU
LSE OPCION #"
 140 LET I$=INKEY$: IF I$="" OR
I$<"1" OR I$>"9" THEN GO TO VAL
"140"
 145 IF I$="8" THEN GO TO VAL "5
"
 150 CLS : GO TO VAL I$*VAL "100
0"
1000 REM INTRODUCCION DE DATOS
1140 CLS : LET P$="EN LA MEMORIA
 CABEN "+STR$ NF+" FICHAS": PRIN
T AT NOT PI,(VAL "32"-LEN P$)/VA
L "2";P$: LET O$="FICHA:"+STR$ N
FI: PRINT AT VAL "2",(VAL "32"-L
EN O$)/VAL "2";O$
1200 PRINT AT VAL "3",NOT PI': L
ET G=Z
1300 INPUT (T$(G));":"; LINE A$:
 IF (A$=" OUT " AND G=PI/PI) THE
N GO TO ME
1350 FOR J=Z TO (H(G)-LEN A$): L
ET A$=A$+" ": NEXT J: IF LEN A$>
H(G) THEN LET A$=A$( TO H(G))
1360 PRINT T$(G);":";A$'
1400 FOR R=Z TO H(G): POKE PUN+R
-Z,CODE A$(R): NEXT R: LET PUN=P
UN+H(G): LET G=G+Z: IF G>T THEN
POKE P2,FN A(PUN): POKE P1,FN B(
PUN,P2): GO TO VAL "1410"
1405 GO TO VAL "1300"
1410 PRINT #NOT PI;AT Z,NOT PI;"
DATOS CORRECTOS ?..............
": LET I$=INKEY$: IF I$="N" THEN
 LET PUN=PUN-NC: GO TO VAL "1140
"
1420 IF I$="S" THEN LET NFI=NFI+
Z: IF NFI>NF THEN GO TO VAL "145
0"
1430 IF I$<>"S" AND I$<>"N" THEN
 GO TO VAL "1410"
1440 GO TO VAL "1140"
1450 PRINT #NOT PI;"NO CABEN MAS
 FICHAS............": PAUSE ME:
 GO TO ME
2000 REM MODIFICACION DE FICHA
2001 LET C$="CORRECCION DE FICHA
............:": GO SUB RO
2002 INPUT "TERMINO A MODIFICAR
? "; LINE M$: LET M$=M$+R$(LEN M
$+Z TO ): FOR A=Z TO T: IF M$=T$
(A) THEN GO TO VAL "2010"
2003 NEXT A
2010 LET N=NC: FOR S=Z TO A-Z: L
ET N=N-H(S): NEXT S: LET FIC=FIC
-N: INPUT (M$+":"); LINE O$
2020 FOR E=LEN O$ TO H(A)-Z: LET
 O$=O$+" ": NEXT E: FOR E=Z TO L
EN O$: POKE (FIC+E),CODE O$(E):
NEXT E
2030 PRINT AT VAL "10",NOT PI;"M
ODIFICACION EFECTUADA": PAUSE ME
: GO TO ME
3000 REM BUSQUEDA DE FICHA
3001 LET C$="BUSQUEDA DE FICHA..
.............:": GO SUB RO
3010 INPUT "PISTA ? "; LINE X$:
FOR W=Z TO LEN X$: IF X$(W)=":"
THEN GO TO VAL "3030"
3020 NEXT W
3025 IF X$(Z)="S" THEN GO TO VAL
 "3800"
3030 LET Q$=X$(W+Z TO ): LET V$=
X$( TO W-Z)+R$(W TO ): FOR Y=Z T
O T: IF V$=T$(Y) THEN GO TO VAL
"3050"
3040 NEXT Y
```

```
3060 POKE D2,FN A(DIR): POKE D1,
FN B(DIR,D2)
3070 LET BT=NOT PI: POKE 65290,F
N A(NC): POKE 65289,FN B(NC,6529
0): FOR I=Z TO Y: LET BT=BT+H(I)
: NEXT I: LET BT=BT-H(I-Z): POKE
 65219,BT: POKE 65218,LEN Q$: FO
R I=Z TO LEN Q$: POKE (DI+I),COD
E Q$(I): NEXT I: POKE 65278,FN A
(DI+Z): POKE 65277,FN B(DI+Z,652
78)
3080 RANDOMIZE USR VAL "65252"
3090 LET FIC=FN C(D1): LET PUN=F
N C(P1): IF FIC>=PUN THEN PRINT
#NOT PI;AT Z,NOT PI;"NO HAY MAS
FICHAS...............": PAUSE ME
: GO TO ME
3100 GO TO VAL "3900"
3800 LET FIC=DIR: FOR S=Z TO NF:
 GO TO 3900
3810 NEXT S: GO TO ME
3900 LET A$="": PRINT AT VAL "3"
,NOT PI;"FICHA: ";INT ((FIC-DIR)
/NC)+Z: LET L=VAL "4": LET FIC=F
IC-Z: FOR I=Z TO T: FOR F=Z TO H
(I): LET A$=A$+CHR$ (PEEK (FIC+F
)): PRINT AT L,NOT PI;T$(I);":";
A$;"█": BEEP .003,20: NEXT F: P
RINT CHR$ VAL "8";" ": LET L=L+Z
+((Z AND LEN A$)VAL "21")*INT (L
EN A$/VAL "21")): IF L>VAL "21"
THEN LET L=VAL "3"
3905 LET A$="": POKE VAL "23692"
,VAL "255": LET FIC=FIC+F-Z: NEX
T I
3910 PRINT #NOT PI;AT NOT PI,NOT
 PI;"MAS FICHAS ?(S/N).........
.....": LET I$=INKEY$: IF I$="S"
 THEN LET FIC=FIC+Z: POKE D2,FN
A(FIC): POKE D1,FN B(FIC,D2): GO
 TO (VAL "3080" AND X$(Z)<>"S")+
(VAL "3810" AND X$(Z)="S")
3920 IF I$="N" THEN GO TO ME
3930 GO TO VAL "3910"
4000 REM BORRAR FICHA
4001 LET C$="BORRADO DE FICHA...
.............:": GO SUB RO
4010 PRINT #NOT PI;AT Z,NOT PI;"
CONFIRMA BORRADO ?(S/N).........
": LET I$=INKEY$: IF I$<>"S" AND
 I$<>"N" THEN GO TO VAL "4010"
4020 IF I$="N" THEN GO TO ME
4025 IF (FIC+1)>=PUN THEN GO TO
VAL "4040"
4030 POKE VAL "65242",FN A(FIC+Z
): POKE VAL "65241",FN B(FIC+Z,V
AL "65242"): POKE VAL "65245",FN
 A(FIC-NC+Z): POKE VAL "65244",F
N B(FIC-NC+Z,VAL "65245"): POKE
VAL "65248",FN A(PUN-FIC+Z): POK
E VAL "65247",FN B(PUN-FIC+Z,VAL
 "65248"): RANDOMIZE USR VAL "65
240"
4040 LET NFI=NFI-Z: LET PUN=PUN-
NC: POKE P2,FN A(PUN): POKE P1,F
N B(PUN,P2)
4050 PRINT AT VAL "10",NOT PI;"F
ICHA SUPRIMIDA": PAUSE ME: GO TO
 ME
5000 REM ORDENACION ALFABETICA
5001 LET C$="ORDENACION ALFABETI
CA..........:": GO SUB RO
5010 INPUT "POR QUE TERMINO ORDE
NO ?........"; LINE Z$: FOR I=Z
TO T: IF Z$=T$(I)( TO LEN Z$) TH
EN GO TO VAL "5030"
5020 NEXT I
5030 POKE 65224,NOT PI: POKE 652
25,NOT PI: POKE 65221,FN A(DIR-Z
): POKE 65220,FN B(DIR-Z,65221):
 POKE 65227,FN A(NFI-Z-Z): POKE
65226,FN B(NFI-Z-Z,65227): POKE
65229,FN A(H(I)): POKE 65228,FN
B(H(I),65229)
5040 LET CT=NOT PI: FOR S=Z TO I
: LET CT=CT+H(S): NEXT S: LET CT
=CT-H(S-Z): POKE 65231,FN A(CT):
 POKE 65230,FN B(CT,65231): POKE
 65239,FN A(PUN-NC): POKE 65238,
FN B(PUN-NC,65239): POKE 65237,F
N A(NC): POKE 65236,FN B(NC,6523
7): LET CO=DIR+CT-Z: POKE 65213,
FN A(CO): POKE 65212,FN B(CO,652
13): LET CO=CO+NC: POKE 65211,FN
 A(CO): POKE 65210,FN B(CO,65211
)
5055 RANDOMIZE USR VAL "65311"
5060 PRINT AT VAL "10",NOT PI;"O
RDENACION EFECTUADA": PAUSE ME:
GO TO ME
6000 REM GRABAR FICHERO
6001 LET C$="GRABACION DE FICHER
O...........:": GO SUB RO
6010 GO SUB VAL "9996": RANDOMIZ
E (PUN-DIR)/NC: SAVE N$CODE VAL
"23670",VAL "2": SAVE N$CODE DIR
,PUN-DIR
6020 PRINT AT VAL "10",NOT PI; F
LASH Z;"VERIFICANDO:";: VERIFY N
$CODE VAL "23670",VAL "2": VERIF
Y N$CODE DIR,PUN-DIR: PRINT "COR
RECTO": PAUSE ME: GO TO ME
7000 REM CARGAR FICHERO
7001 LET C$="CARGA DE FICHERO...
...........:": GO SUB RO
7010 GO SUB VAL "9996": LOAD N$C
ODE : LOAD N$CODE DIR: LET PUN=D
IR+NC*FN C(VAL "23670"): POKE P2
,FN A(PUN): POKE P1,FN B(PUN,P2)
: PRINT AT VAL "10",NOT PI;"CORR
ECTO": PAUSE ME: LET NFI=INT ((P
UN-DIR)/NC)+Z: GO TO ME
9000 REM IMPRESORA
9001 LET C$="LISTADO EN IMPRESOR
A............": GO SUB RO
9005 PRINT "PULSE UNA TECLA PARA
 VOLVER AL  MENU"
9010 INPUT "DESDE LA FICHA: ";F1
: IF F1=NOT PI THEN LET F2=NFI-Z
: GO TO VAL "9020"
9015 INPUT "HASTA LA FICHA: ";F2
9030 FOR F=F1+(Z AND F1=NOT PI)
TO F2: LPRINT "FICHA: ";F: FOR G
=Z TO T: LET O=NOT PI: LET A$=""
9031 FOR X=Z TO G: LET O=O+H(X):
 NEXT X: LET O=O-H(X-Z)
9035 LPRINT T$(G);: FOR J=Z TO H
(G): LET A$=CHR$ PEEK (F*NC+DIR+
J-Z-NC+O): LPRINT A$;: NEXT J: L
PRINT ': NEXT G: LPRINT "-------
-------------------------": IF I
NKEY$="" THEN GO TO VAL "9050"
9040 LET A$="": GO TO ME
9050 LPRINT ': NEXT F: LET A$=""
: GO TO ME
9996 INPUT "NOMBRE DEL FICHERO ?
 "; LINE N$: IF LEN N$>VAL "10"
THEN GO TO VAL "9996"
9997 RETURN
9998 PRINT AT Z,NOT PI; INK VAL
"6"; PAPER Z; BRIGHT Z;C$: RETUR
N
9999 SAVE "ficheros" LINE NOT PI
: VERIFY "ficheros"
```

LISTADO ENSAMBLADOR DEL PROGRAMA FICHERO

```
RUTINAS C/M PROGRAMA FICHEROS:

65240 2147AE              LD    HL,446
15
65243 11E3AD              LD    DE,445
15
65246 010100              LD    BC,000
01
65249 EDB0                LDIR
65251 C9                  RET
65252 3ABFFE       LFEE4  LD    A,(652
15)
65255 67                  LD    H,A
65256 3AC1FE              LD    A,(652
17)
65259 BC                  CP    H
65260 F8                  RET   M
65261 3ABEFE              LD    A,(652
14)
65264 6F                  LD    L,A
65265 3AC2FE              LD    A,(652
18)
65268 47                  LD    B,A
65269 3AC3FE              LD    A,(652
19)
65272 5F                  LD    E,A
65273 1600                LD    D,#00
65275 19                  ADD   HL,DE
65276 110E5B              LD    DE,233
10
65279 1A           LFEFF  LD    A,(DE)
65280 BE                  CP    (HL)
65281 2005                JR    NZ,LFF
08
65283 23                  INC   HL
65284 13                  INC   DE
65285 10F8                DJNZ  LFEFF
65287 C9                  RET
65288 116400       LFF08  LD    DE,001
00
65291 3ABEFE              LD    A,(652
14)
65294 6F                  LD    L,A
65295 3ABFFE              LD    A,(652
15)
65298 67                  LD    H,A
65299 19                  ADD   HL,DE
65300 7D                  LD    A,L
65301 32BEFE              LD    (65214
),A
65304 7C                  LD    A,H
65305 32BFFE              LD    (65215
),A
65308 18C6                JR    LFEE4
65310 00                  NOP
65311 2AC4FE       LFF1F  LD    HL,(65
220)
65314 22C6FE              LD    (65222
),HL
65317 2ABCFE              LD    HL,(65
212)
65320 22D0FE              LD    (65232
),HL
65323 2ABAFE              LD    HL,(65
210)
65326 22D2FE              LD    (65234
),HL
65329 2ACAFE              LD    HL,(65
226)
65332 ED4BC8FE            LD    BC,(65
224)
```

```
65336 03                  INC   BC
65337 ED43C8FE            LD    (65224
),BC
65341 B7                  OR    A
65342 ED42                SBC   HL,BC
65344 F8                  RET   M
65345 010000       LFF41  LD    BC,000
00
65348 03           LFF44  INC   BC
65349 2ACCFE              LD    HL,(65
228)
65352 B7                  OR    A
65353 ED42                SBC   HL,BC
65355 FA87FF              JP    M,LFF8
7
65358 2AD0FE              LD    HL,(65
232)
65361 09                  ADD   HL,BC
65362 56                  LD    D,(HL)
65363 2AD2FE              LD    HL,(65
234)
65366 09                  ADD   HL,BC
65367 7E                  LD    A,(HL)
65368 BA                  CP    D
65369 28E9                JR    Z,LFF4
4
65371 F287FF              JP    P,LFF8
7
65374 010100              LD    BC,000
01
65377 2AD0FE       LFF61  LD    HL,(65
232)
65380 ED5BCEFE            LD    DE,(65
230)
65384 B7                  OR    A
65385 ED52                SBC   HL,DE
65387 09                  ADD   HL,BC
65388 7E                  LD    A,(HL)
65389 ED5BD4FE            LD    DE,(65
236)
65393 19                  ADD   HL,DE
65394 56                  LD    D,(HL)
65395 77                  LD    (HL),A
65396 B7                  OR    A
65397 7A                  LD    A,D
65398 ED5BD4FE            LD    DE,(65
236)
65402 ED52                SBC   HL,DE
65404 77                  LD    (HL),A
65405 03                  INC   BC
65406 EB                  EX    DE,HL
65407 B7                  OR    A
65408 ED42                SBC   HL,BC
65410 FA87FF              JP    M,LFF8
7
65413 18DA                JR    LFF61
65415 2AC6FE       LFF87  LD    HL,(65
222)
65418 ED5BD4FE            LD    DE,(65
236)
65422 19                  ADD   HL,DE
65423 22C6FE              LD    (65222
),HL
65426 2AD0FE              LD    HL,(65
232)
65429 19                  ADD   HL,DE
65430 22D0FE              LD    (65232
),HL
65433 2AD2FE              LD    HL,(65
234)
65436 19                  ADD   HL,DE
65437 22D2FE              LD    (65234
),HL
65440 ED5BD6FE            LD    DE,(65
238)
65444 2AC6FE              LD    HL,(65
222)
65447 23                  INC   HL
65448 B7                  OR    A
65449 ED52                SBC   HL,DE
65451 FA41FF              JP    M,LFF4
1
65454 C31FFF              JP    LFF1F
```

GUN FIRE

LIST

```
  10 REM POR LUIS JORGE GARCIA
  20 REM  1987.MUNDO SPECTRUM
  30 POKE 23607,163: POKE 23658,
0: BRIGHT 1: PAPER 0: INK 2: BOR
DER 0: CLEAR 44999: LET REC=100:
 LET Z$="LUI"
  40 GO SUB 170
  50 GO SUB 280
  60 GO SUB 360
  70 LET VE=11: LET H=4: LET PU=
0: LET HE=25: LET XX=25: LET LE=
1: LET BA=4: LET VEE=12: LET ZZ=
12
  80 GO SUB 390
  90 GO SUB 490
 100 GO SUB 490
 110 GO SUB 470
 120 GO SUB 620
 130 GO SUB 700
 140 IF LE=5 THEN GO SUB 620
 150 GO TO 90
 160 REM P
 170 LET L$="GUNFIRE.": LET P=70
: LET A=LEN L$: PRINT INK 0;AT 2
1,0;L$
 180 FOR N=0 TO 8*A-1: FOR O=0 T
O 7
 190 IF POINT (N,O)=0 THEN GO TO
 220
 200 BEEP .001,60
 210 PLOT N*4,O*4+135-P: DRAW 4,
0: DRAW 0,4: DRAW -4,0: DRAW 0,-
3: DRAW 3,0: DRAW 0,2: DRAW -2,0
: DRAW 0,-1: DRAW 2,0: DRAW -2,-
2
 220 NEXT O: NEXT N
 230 PRINT AT 15,4; INK 7;"POR L
UIS JORGE GARCIA";AT 21,0; INK 7
;"                ": POKE 23692,100: FOR
 N=0 TO 9: PRINT : NEXT N: PRINT
 #1;" DEJA PULSADA 'S' PARA EL M
ENU"
 240 RESTORE 240: FOR N=64400 TO
 64424: READ A: POKE N,A: NEXT N
 250 DATA 6,64,17,0,64,213,225,3
5,197,1,31,0,26,237,176,43,119,0
,35,35,19,193,16,240,201
 260 FOR F=0 TO 31: RANDOMIZE US
R 64400: PAUSE 5: NEXT F: IF INK
EY$="S" THEN RETURN
 270 GO TO 260
 280 REM R
 290 FOR N=0 TO 60: BEEP .01,N:
NEXT N: INPUT "TECLA ABAJO. "; L
INE D$
 300 INPUT "TECLA ARRIBA. "; LIN
E I$
 310 INPUT "TECLA DISPARO. "; LI
NE F$
 320 PRINT AT 10,4; INK 7; PAPER
 2; BRIGHT 1; FLASH 1;"ESTA BIEN
 ASI (S/N)?."
 330 PRINT AT 13,5; INK 7;"TECLA
 ABAJO.          ";D$;AT 15,5;"TECLA
 ARRIBA.         ";I$;AT 17,5;"TECLA
 DISPARO.        ";F$
 340 INPUT LINE C$: IF C$="N" TH
EN GO TO 280
 350 RETURN
 360 REM M
 370 RESTORE 370: FOR F=1 TO 53:
 READ D,T: BEEP D,T: NEXT F: DAT
A .1,5,.1,5,.1,5,.4,7,.4,7,.4,7,
.1,69,.1,7,.1,7,.1,5,.2,7,.4,10,
```

Tu misión será interceptar las naves enemigas, antes de que estas lleguen hasta tí. De lo contrario, te quitarán una vida. (Y solo tienes cuatro).

Podrás conseguir un bonus, en puntos, al llegar a los 1000 puntos justo, es decir, no podrás llegar por ejemplo con 1005, ya que entonces no se te darán bonus.

A los 100, 300, 600 y 3500 la dificultad irá incrementando y te será más difícil conservar la vida.

¡SUERTE Y AL TORO!

LAS MEJORES PUNTUACIONES ADMITIRAN REGISTRAR EL NOMBRE DEL JUGADOR.



Líneas 160 a 230
Realización del rótulo GUNFIRE.

Líneas 240 a 260
Realización de un scroll en el 1. tercio de pantalla.

Líneas 280 a 350
Redefinición de teclas.

Líneas 360 a 380
Realización de la música.

Líneas 390 a 480
Iniciación de Pantalla.

Líneas 490 a 610
Movimiento de la pistola, niveles de dificultad y suma de bonus.

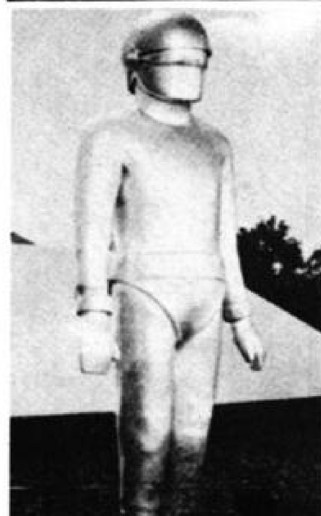
Líneas 620 a 690
Generador de enemigos n.1.

Líneas 700 a 740
Generador de enemigos n.2.

Líneas 750 a 810
Cartelera Game Over, efecto musical y inicializador de juego.

Líneas 820 a 910
Códigos chr$ retocados y gráficos. (Esta línea de 820 a 910 se puden anular pero no podrás jugar con gráficos y las letras serán las típicas del spectrum).

Rec.	Record máxima puntuación.
Z$.	Record realizado por "......".
VE .	Coord. vertical de la nave.
H.	Coord. horizontal de la nave.
PU.	Puntos realizados.
HE.	Coord. horizontal de enemigo 1. BOM (Borrado del enemigo 1)
VEE.	Coord. vertical de la nave enemiga. VEH (Borrado del enemigo 1).
XX.	Coord. horizontal del enemigo 2.
ZZ.	Coord. vertical del enemigo 2.
BA.	Nº de vidas.
LE.	Nivel.
L$.	Rótulo. (Nombre del juego).
D$.	Variable que guarda el valor de la tecla ABAJO.
I$.	Variable que guarda el valor de la tecla ARRIBA.
F$.	Variable que guarda el valor de la tecla DISPARO.



AVISO IMPORTANTE

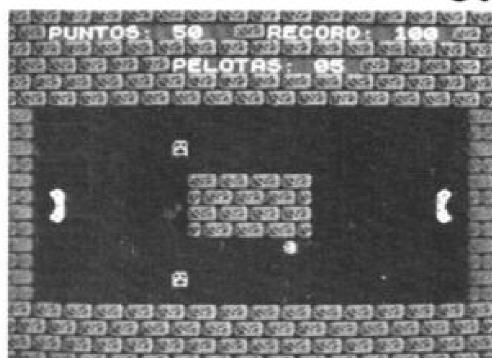
Copiar con detención las líneas datas, reparándolas con detenimiento, ya que de lo contrario los gráficos y letras podrían salir corrompidas.

Poner espacios donde vienen indicados.

```
.1,7,.1,5,.1,7,.1,5,.2,4,.1,69,.
1,4,.1,5,.1,4,.4,7,.4,7,.4,7,.1,
69,.1,7,.1,7,.1,5,.2,7,.4,10,.1,
7,.1,5,.1,7,.1,5,.2,4,.1,69,.1,1
2,.1,12,.1,12,.1,12,.1,11,.1,10,
.2,9,.1,12,.1,12,.1,12,.1,14,.1,
13,.1,12,.2,11,.1,4,.1,5,.2,4,.1
,7,.3,0,.1,3,.3,5,.2,3,.2,69
 380 RETURN
 390 REM I
 400 LET  =7
 410 PRINT AT  ,0; INK 7; PAPER
0;"
  ": LET  = +1
 420 IF  < 22 THEN GO TO 410
 430 BORDER 0: INK 7: PLOT 24,11
0: DRAW 192,0: DRAW 0,-71: DRAW
-192,0: DRAW 0,71: PLOT 24,39: D
RAW 192,0: DRAW 0,-39: DRAW -192
,0: DRAW 0,39
 440 PRINT #1; BRIGHT 1; PAPER 2
;"MUNDO SPECTRUM....MUNDO SPECTR
UM"
 450 PRINT AT 18,4;"VIDAS:";AT 1
8,16;"MAX:";REC
 460 FOR N=0 TO 3: PRINT AT 18,1
1+N;"Ӿ": BEEP .002,N: NEXT N
 470 PRINT AT 20,4;"PUNTOS:";PU;
 PRINT AT 20,16;"RECORD:";Z$
 480 RETURN
 490 REM M
 500 IF INKEY$=D$ THEN LET VE=VE
+1: BEEP .001,30: PRINT AT VE-1,
H;" "
 510 IF INKEY$=I$ THEN LET VE=VE
-1: BEEP .001,37: PRINT AT VE+1,
H;" "
 520 IF VE>15 THEN LET VE=15
 530 IF VE<8 THEN LET VE=8
 540 IF INKEY$=F$ THEN FOR F=1 T
O 20 : PRINT AT VE,(H+1)+F; INK
6;"-": NEXT F: BEEP .001,69: BEE
P .002,40: PRINT AT VE,H+1;"
                       ": GO TO 790
 550 IF PU=100 THEN LET LE=2
 560 IF PU=300 THEN LET LE=3
 570 IF PU=600 THEN LET LE=4
 580 IF PU=1000 THEN LET LE=5: L
ET BON=200: FOR N=1 TO 40: PRINT
 AT 11,9; FLASH 1;"BONUS. ";BON:
 BEEP .0004,40: LET PU=PU+5: PRI
NT AT 20,11;PU: NEXT N: PRINT AT
 11,9; FLASH 0;"          "
 590 IF PU=3500 THEN LET LE=7
 600 PRINT AT VE,H;"Ӿ"
 610 RETURN
 620 REM G                       1
 630 LET BOH=HE: LET VEH=VEE
 640 LET VEE=INT (RND*LE)+8
 650 PRINT AT VEH,BOH+1;" "
 660 PRINT AT VEE,HE; INK 7;"♠"
 670 LET HE=HE-1
 680 IF HE=4 THEN BEEP .1,25: LE
T BA=BA-1: PRINT AT 18,11+BA;" "
;AT VEE,HE+1;" ": LET HE=25: IF
BA=0 THEN GO TO 750
 690 RETURN
 700 REM                         2
 710 PRINT AT ZZ,XX; INK 2;"◎ "
 720 LET XX=XX-1
 730 IF XX=4 THEN BEEP .1,45: LE
T BA=BA-1: PRINT AT 18,11+BA;" "
;AT ZZ,XX+1;" ": LET XX=25: LET
ZZ=INT (RND*8)+8: IF BA=0 THEN G
O TO 750
 740 RETURN
 750 REM S
 760 CLS : PRINT AT 11,10; FLASH
 1; INK 7; PAPER 2;"GAME OVER":
PAUSE 100: FOR N=0 TO 42 STEP 3:
 FOR M=0 TO 17 STEP 3: BEEP .02,
M+N: BEEP .007,M-N: NEXT M: NEXT
 N: PAUSE 100
 770 IF PU>=REC THEN LET REC=PU:
 PRINT AT 21,0;" RECORD INTRODUC
E MAX.3 LETRAS. ": INPUT "NOMBRE
. "; LINE Z$: IF LEN Z$>3 THEN G
O TO 750
 780 PAPER 0: INK 2: CLS : GO TO
 40
 790 IF VE=VEE THEN BEEP .01,7:
LET PU=PU+10: LET HE=25
 800 IF VE=ZZ THEN BEEP .01,9: L
ET PU=PU-5: LET XX=25: LET ZZ=IN
T (RND*8)+8
 810 GO TO 550
 820 PAPER 0: INK 2: BORDER 0: C
LS : PRINT #1; FLASH 1; BRIGHT 1
; PAPER 2; INK 7;"CARGANDO DATAS
 ESPERA UN MOMENTO"
 830 RESTORE 830: FOR N=41984 TO
 42487: READ A: POKE N,A: NEXT N
 840 DATA 0,0,0,0,0,0,0,0,24,24,
24,24,24,0,24,0,54,54,54,0,0,0,0
,0,108,254,254,108,254,254,108,0
,24,254,152,254,26,254,24,0,198,
198,12,24,48,102,198,0,24,38,24,
48,74,70,62,0,8,8,16,0,0,0,0,0
 850 DATA 4,8,8,8,8,8,4,0,32,16,
16,16,16,16,32,0,0,84,56,254,56,
84,0,0,0,8,8,62,8,8,0,0,0,0,0,0,
0,8,8,16,0,0,0,60,0,0,0,0,0,0,0,
0,24,24,0,0,2,4,8,16,32,64,128,0
,254,254,206,214
 860 DATA 230,254,254,0,48,112,2
40,48,48,252,252,0,254,254,2,254
,192,254,254,0,254,254,6,62,6,25
4,254,0,192,204,204,252,252,12,1
2,0,254,254,192,254,6,254,254,0,
254,254,128,254,198
 870 DATA 254,254,0,254,254,6,6,
6,6,6,0,254,254,198,254,198,254,
254,0,254,254,198,254,6,254,254,
0,0,24,24,0,24,24,0,0,0,24,24,0,
24,8,16,0,16,32,64,128,64
 880 DATA 32,16,0,0,60,0,60,0,0,
0,0,4,2,1,2,4,8,0,24,36,4,8,16
,0,16,0,60,130,154,148,152,130,1
24,0,254,254,198,254,254,198,198
,0,254,254,198,252,198,254,254,0
,254,254
 890 DATA 198,192,198,254,254,0,
240,252,198,198,198,252,240,0,25
4,254,194,240,194,254,254,0,254,
254,194,240,192,192,192,0,254,25
4,192,206,198,254,254,0,198,198,
254,254,198,198,198,0,126,126,24
,24,24,126,126,0,254,254,6,198,1
98,254,254,0,206
 900 DATA 216,240,240,216,204,19
8,0,192,192,192,192,198,254,254,
0,198,230,254,214,198,198,198,0,
198,230,246,222,206,198,198,0,25
4,254,198,198,198,254,254,0,254,
254,198,254,254,192,192,0,254,25
4,198,198,206,254,2,0,254,254,19
4,254,252,204,198,0,254
 910 DATA 254,192,254,2,254,254,
0,254,254,24,24,24,24,24,0,198,1
98,198,198,198,254,254,0,198,198
,198,198,108,56,16,0,198,198,198
,214,254,238,198,0,134,204,120,5
6,108,198,130,0,198,198,254,254,
24,24,24,0,254,254,12,24,48,126,
254,0,254
 920 DATA 60,152,255,255,152,60,
254,0,0,0,111,111,0,0,0,24,60,90
,255,36,24,36,66,153,60,66,219,2
19,66,60,153
 930 INPUT "QUIERES SACAR UNA CO
PIA?.";A$: IF A$="S" OR A$=" " T
HEN SAVE "GUNFIRE" LINE 960: SAV
E "GUNLETER"CODE 41992,496: RUN
 940 RUN
 950 SAVE "GUNFIRE" LINE 820: ST
OP
 960 CLS : LOAD ""CODE : RUN
```

BI PALETA





Hoy en día consideramos el espacio como algo que se encuentra fuera de los dominios de la raza humana, pero eso no quita, que existan seres intergalácticos en otras galaxias distintas a la Vía Lactea. Pues bien, en este juego podrás ver la manera de jugar al tenis que tienen estos seres extraterrestres, que junto con dos raquetas manejan una bola muy especial, con la cual se deberá destruir los pequeños fantasmillas que rondan dentro del campo donde se debuta el campeonato de ping-pong.

Cada fantasma que consigas destruir te puntuará en el marcador tanto como 50 puntos, los cuales cuando vayan sumándose, te harán superar el record de la máquina. Suerte y a por todas.

Las teclas de movimiento son las siguientes: con la Q o la P mueves hacia arriba. Con la Z o la M mueves hacia abajo, y con el espacio sacas la pelota.

Comentario del programa

En la línea 40.- se inicializan los colores de la pantalla, a la vez que se añade un cartel en medio donde se hace visualizar -Espera un momento-. Ese momento de espera es debido a lo que tarda el ordenador en introducir los UDGs. del programa, en memoria, metidos estos últimos mediante datas, los cuales son leídos gracias a la línea 45.

En la línea 50.- se inicializa la variable del record igualandola a 100, y se realiza un poke especial con el cual se asegura que el cursor se encuentre durante el juego en mayusculas.

En la línea 60.- se efectúa un CLS en la pantalla, y se la vuelve a re-inicializar en sus colores, que perdurarán en el juego.

Desde la línea 70 a la línea 150.- se inicializan todas las variables del juego, exceptuando la del record que ha sido inicialiada anteriormente. Las variables se corresponden de la siguiente manera: BORRAR.- variable que se utiliza para borrar la bola o las raquetas sólo cuando se encuentren en movimiento, de esta manera al estar paradas no se notará ningún parpadeo. PU.- variable de

```
  10 REM ‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡
  20 REM ‡‡‡  BI-PALETA  ‡‡‡
  30 REM ‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡
  35 REM
  36 REM
  40 PAPER 0: BORDER 0: INK 7: B
RIGHT 1: CLS : PRINT PAPER 2;AT
10,7;" ESPERA UN MOMENTO ";AT 9,
7;"                 ";AT 11,7;
"                 "
  45 FOR A=0 TO 87: READ B: POKE
 USR "a"+A,B: NEXT A
  50 LET RE=100: POKE 23658,8
  60 BRIGHT 0: BORDER 0: PAPER 0
: INK 7: CLS
  70 REM ‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡
  80 REM ‡‡‡‡  VARIABLES  ‡‡‡‡
  90 REM ‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡
 100 LET BORRAR=1: LET PU=0
 110 LET PX=10: LET PXB=PX
 120 LET FYI=1: LET FE=0: LET FX
=16: LET FY=15: LET FXB=FX: LET
FYB=FY
 140 LET CBO=5: LET BOLA=0: LET
BX=14: LET BY=3: LET BXB=BX: LET
 BYB=BY: LET BXI=1: LET BYI=1
 150 LET MOV=1
 200 REM ‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡
 210 REM ‡‡‡ DIBUJO PANTALLA ‡‡‡
 220 REM ‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡‡
 230 PRINT INK 4;AT 0,0;"JKJKJKJ
KJKJKJKJKJKJKJKJKJKJKJKJKIJKJKJK
JKJKJKJKJKJKJKJKJKJKJKJKIJKJKJKJ
KJKJKJKJKJKJKJKJKJKJKJKJKIJKJKJK
```

BI-PALETA



los puntos. PX.- variable de movimiento de las dos raquetas. PXB.- variable de borrado de las raquetas. FYI.- variable de incremento o decremento de los fantasmas según lo necesite su movimiento.



FE.- variable de si existe o no los fantasmas en pantalla. FX,FY.- variables de movimiento de los dos fantasmas. FXB, FYB.- variables de borrado de los dos fantasmas. CBO.- contador de las bolas. BOLA.- variable de existe o no la bola o pelota. BX, BY.- coordenadas de movimiento de la pelota. BXB, BYB.- coordenadas de borrado de la pelota. BXI, BYI.- coordenadas de incremento o decremento de la pelota. MOV.- contador de la pelota para que a cada gráfico o movimiento de esta aparezca un gráfico diferente de esta.

Desde la línea 200 a la línea 270.- se dibuja el paisaje y contorno de la pantalla donde se va a desarrollar el juego.

Desde la línea 1000 a la línea 1400.- se encuentra el bucle principal donde se imprime y mueve mediante las instrucciones de INKEY$, las raquetas, a la vez que se realizan los correspondientes GOSUB a las distintas subrutinas del programa.

Desde la línea 1500 a la línea 1580.- se imprimen y

A	B	C	D	E	F	G	H	I	J	K

```
JKJKJKJKJKJKJKJKJKJKJKJKIJKJKJKJ
KJKJKJKJKJKJKJKJKJKJKJKIJKJKJK
JKJKJKJKJKJKJKJKJKJKJKJKI"
 240 PRINT INK 4;AT 18,0;"JKJKJK
JKJKJKJKJKJKJKJKJKJKJKJKIJKJKJ
KJKJKJKJKJKJKJKJKJKJKJKIJKJKJK
JKJKJKJKJKJKJKJKJKJKJKJKIJKJKJ
KJKJKJKJKJKJKJKJKJKJKJKI"
 250 PRINT INK 4;AT 10,12;"JKJKJ
KJK";AT 11,12;"IJKJKJKI";AT 12,1
2;"JKJKJKJK";AT 13,12;"IJKJKJKI"
 260 FOR A=6 TO 16 STEP 2: PRINT
 INK 3;AT A+1,0;"II";AT A+1,30;"
II";AT A,0;"JK";AT A,30;"JK": NE
XT A
 270 PRINT BRIGHT 1; PAPER 1;AT
1,3;"PUNTOS: 00  ";AT 1,17;"RECO
RD:     ";AT 1,25;RE;AT 3,11;"PE
LOTAS: 05 "
1000 REM ***********************
1010 REM ** IMPRESION PALETAS **
1020 REM ***********************
1030 IF BORRAR=1 THEN PRINT AT P
XB,3;" ";AT PXB+1,3;" ";AT PXB,2
8;" ";AT PXB+1,28;" ": LET BORRA
R=0
1040 LET PXB=PX
1050 PRINT BRIGHT 1;AT PX,3;"A";
AT PX+1,3;"B";AT PX,28;"C";AT PX
+1,28;"D"
1060 IF INKEY$="Q" OR INKEY$="P"
 THEN LET PX=PX-1: LET BORRAR=1:
 IF PX<6 THEN LET PX=6
1070 IF INKEY$="Z" OR INKEY$="M"
 THEN LET PX=PX+1: LET BORRAR=1:
 IF PX>16 THEN LET PX=16
1100 IF FE=0 THEN LET FE=1: LET
FX=16: LET FY=15
1110 IF FE=1 THEN GO SUB 1500
1140 IF BOLA=0 THEN LET BX=PX+1:
 LET BY=4: GO SUB 2120: IF INKEY
$=" " THEN LET BOLA=1: LET BXI=1
: LET BYI=1: FOR A=1 TO 5: BEEP
.005,A: NEXT A
1150 IF BOLA=1 THEN GO SUB 2000
1400 GO TO 1030
1500 REM ***********************
1510 REM ** MOVIM. FANTASMA 1 **
1520 REM ***********************
1530 PRINT AT FXB,FYB;" ";AT FXB
-8,FYB;" ": LET FXB=FX: LET FYB=
FY
1540 PRINT INK 6;AT FX,FY;"E";AT
 FX-8,FY;"E"
1550 LET FY=FY+FYI
1560 IF FY>25 THEN LET FYI=-1
1570 IF FY<6 THEN LET FYI=1
1580 RETURN
2000 REM *********************
2010 REM ** MOVIMIENTO BOLA **
2020 REM *********************
2120 PRINT AT BXB,BYB;" ": LET B
XB=BX: LET BYB=BY
2130 IF MOV=1 THEN PRINT INK 5;A
T BX,BY;"H"
2140 IF MOV=2 THEN PRINT INK 5;A
T BX,BY;"G"
2145 IF BOLA<>0 THEN LET MOV=3-M
OV
2150 IF BOLA=0 THEN LET MOV=1: R
ETURN
2152 IF BXI=1 AND BYI=1 AND ATTR
 (BX,BY+1)=4 THEN LET BYI=-1: GO
 TO 2200
2154 IF BXI=-1 AND BYI=1 AND ATT
```

```
R (BX-1,BY)=4 THEN LET BXI=1: GO
 TO 2200
2156 IF BXI=1 AND BYI=-1 AND ATT
R (BX+1,BY)=4 THEN LET BXI=-1: G
O TO 2200
2158 IF BXI=-1 AND BYI=-1 AND AT
TR (BX,BY-1)=4 THEN LET BYI=1: G
O TO 2200
2160 IF ATTR (BX+BXI,BY+BYI)=4 A
ND BXI=1 AND BYI=1 THEN LET BXI=
-1: GO TO 2200
2170 IF ATTR (BX+BXI,BY+BYI)=4 A
ND BXI=-1 AND BYI=1 THEN LET BYI
=-1: GO TO 2200
2180 IF ATTR (BX+BXI,BY+BYI)=4 A
ND BXI=1 AND BYI=-1 THEN LET BYI
=1: GO TO 2200
2190 IF ATTR (BX+BXI,BY+BYI)=4 A
ND BXI=-1 AND BYI=-1 THEN LET BX
I=1
2192 IF ATTR (BX+BXI,BY+BYI)=3 T
HEN GO SUB 3500
2194 IF ATTR (BX+BXI,BY+BYI)=6 T
HEN GO SUB 3000
2196 IF ATTR (BX+BXI,BY+BYI)=6 T
HEN GO SUB 3000
2198 IF ATTR (BX+BXI,BY+BYI)=71
THEN GO SUB 3200
2200 IF BOLA<>0 THEN LET BX=BX+B
XI: LET BY=BY+BYI
2210 RETURN
3000 REM ***********************
3010 REM ** CHOQUE FANTASMAS **
3020 REM ***********************
3030 FOR A=1 TO 10
3040 PRINT INK INT (RND*8);AT BX
B+BXI,BYB+BYI;"E"
3050 BEEP .009,0: BEEP .006,5
3060 NEXT A
3070 PRINT AT BXB+BXI,BYB+BYI;"
"
3080 LET PU=PU+50: PRINT PAPER 1
; BRIGHT 1;AT 1,11;PU
3090 PRINT AT FXB,FYB;" ";AT FXB
-8,FYB;" ": LET FY=15
3100 RETURN
3200 REM ***********************
3210 REM *** CHOQUE RAQUETA ***
3220 REM ***********************
3230 IF BXI=1 AND BYI=1 THEN LET
 BXI=-1: RETURN
3240 IF BXI=-1 AND BYI=1 THEN LE
T BYI=-1: RETURN
3250 IF BXI=1 AND BYI=-1 THEN LE
T BYI=1: RETURN
3260 IF BXI=-1 AND BYI=-1 THEN L
ET BXI=1
3270 RETURN
3500 REM ***********************
3510 REM ** CONTADOR PELOTAS **
3520 REM ***********************
3530 FOR A=1 TO 20: PRINT INK IN
T (RND*8);AT BXB,BYB;"E": BEEP .
005,A: BEEP .005,0 : NEXT A
3540 PRINT AT BXB,BYB;" "
3550 LET BOLA=0: LET CBO=CBO-1
3555 PRINT PAPER 1; BRIGHT 1;AT
3,21;CBO
3560 IF CBO=0 THEN GO TO 4000
3570 RETURN
4000 REM ***********************
4010 REM *** FINAL PROGRAMA ***
4020 REM ***********************
4030 IF PU>RE THEN LET RE=PU
4040 PRINT INK INT (RND*8);AT 11
,14;"GAME";AT 12,14;"OVER"
```

BI-PALETA



mueven los dos fantasmas en pantalla.

Desde la línea 2000 a la línea 2210.- se realiza la impresión y el movimiento de la bola en la pantalla, a la vez que desde la línea 2152 a la línea 2200 están todas las comprobaciones de los choques contra la pelota, tanto de raqueta, fantasmas o contra la pared.

Desde la línea 3000 a la línea 3100.- se realiza lo que pasaría en pantalla, si choca la pelota contra los fantasmas.

Desde la línea 3200 a la línea 3270.- se efectua el incremento o decremento de la pelota según choque esta contra la raqueta.

Desde la línea 3500 a la línea 3570.- se imprimen las pelotas en pantalla (su número), a la vez que se van contabilizando.

Desde la línea 4000 a la línea 4080.- se realiza el final del programa, donde aparece el clásico GAME OVER en pantalla.

Y por último, desde la línea 9000 a la línea 9080.aparecen todos los Datas de los correspondientes UDGs del programa.

```
4045 PRINT BRIGHT 1;AT 15,9;"OTR
A VEZ (S/N)
4050 BEEP .005,10: BEEP .004,0:
BEEP .006,5
4060 IF INKEY$="S" THEN GO TO 60
4070 IF INKEY$="N" THEN STOP
4080 GO TO 4040
9000 REM ********************
9010 REM *** DATAS UGDs. ***
9020 REM ********************
9030 DATA 28,126,103,53,31,29,13
,11,11,13,29,31,53,103,126,28
9040 DATA 56,126,230,172,248,184
,176,208,240,176,184,248,172,230
,126,56
9050 DATA 126,219,153,255,231,19
5,255,165,0,71,22,48,180,2,198,2
08
9060 DATA 60,94,110,24,24,126,11
0,60,0,60,94,110,126,110,60,0
9070 DATA 124,254,182,218,174,22
2,124,0,127,255,183,219,174,215,
127,0
9080 DATA 252,222,126,238,222,25
4,252,0
```

PROXIMAS SECCIONES



CORREO DEL LECTOR

Consultas, dudas, problemas sugerencias, etc. de los lectores que estén relacionados con el sistema SPECTRUM, serán publicadas y respondidas en esta sección.

CLUB

En este apartado daremos cabida a todos los anuncios entre particulares o clubs de informática que estén interesados en cambiar, comprar o vender algo, establecer contactos, etc

Enviad vuestros mensajes: Mundo SPECTRUM, Tomás López, 3 6.º 28009 MADRID

DISCIPLE



Interface Centronic de impresora incorporado.

List y copy de programas con y sin grises.

Doble interface de joystick convencional Sinclair/Kempston.

1 ó 2 unidades de disco 3 1/2" ó 5 1/4".

800 Kbs. por disco. Alta velocidad de transferencia.

No ocupa RAM.

Transfer de programas 48/128 Kbs. y pantallas.

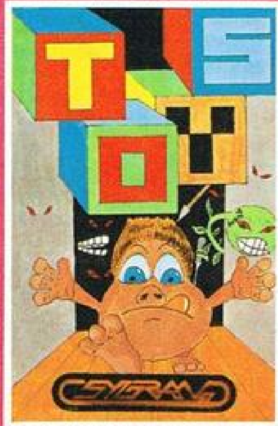
Totalmente compatible con el software existente para Microdrive.

Ficheros secuenciales y acceso aleatorio.

TEENEX

Ayala, 86
Teléfs. 435 64 20
28001 MADRID

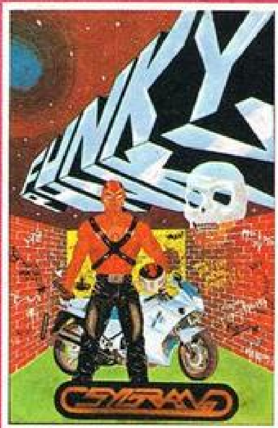
PARA TU...



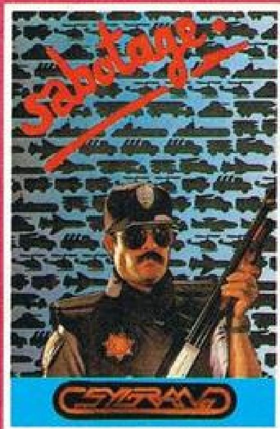
AMSTRAD



SPECTRUM



MSX



MSX

500 pelas